ON A CLASS OF STOCHASTIC APPROXIMATION-TYPE PARAMETER-LEARNING ALGORITHMS FOR PATTERN RECOGNITION

AMITA PAL (née PATHAK)
Electronics and Communication Sciences Unit
Indian Statistical Institute
Calcutta
India

Thesis submitted to the Indian Statistical Institute in partial fulfilment of the requirements for the degree of DOCTOR OF PHILOSOPHY, 1990

Acknowledgements

I take great pleasure in acknowledging my sincere gratitude and indebtedness to all those who contributed in diverse ways to making this thesis a reality, particularly to:

- the Indian Statistical Institute, for making a variety of resources available for this work, and for providing an environment congenial to research,
- Dr. Dwijesh K. Dutta Majumder, my supervisor, for the great interest he has always taken in my work and for his guidance and his constructive criticism,
- Dr. Sankar K. Pal, for having suggested this line of research in the first place, for having supervised my work for about four and a half years, and for his constant encouragement, criticism and support,
- Mr. Chandranath Basu and Mr. S. Sai Giridhar, for their immense contribution to the programming aspects of this work,
- Mr. Niranjan Chatterjee, for having done the typographical work with such care and with great attention to detail,
- Mr. Sunil Chakraborty, for drawing all the figures in this work,
- Mr. R. N. Kar of the Computing and Statistical Services Centre of the Indian Statistical Institute, for helping with laser printer,
- the Reprography Unit of the Indian Statistical Institute, for taking care of all the reprographic work involved in the preparation of the thesis,
- the Binding Unit of the Indian Statistical Institute, for binding all the copies of the thesis,
- Mr. N. R. Pal, for having helped in procuring the paper required for printing the thesis,
- Ms Swati Choudhury, for helping me in the tedious task of proofreading,
- all colleagues at the Electronics and Communication Sciences Unit of the Indian Statistical Institute, notably, Mr. M. K. Kundu, Mr. Swapan Seal and Mr. S. E. Sharma, for helping in various ways,
- my friends Ms Basabi Chakraborty, Ms Swati Choudhury, Ms Anjana Dewanjee (née Banerjee) and Mr. Anup K. De for their encouragement and for making life more pleasant at the institute during those uncertain years,
- my family, for their patience and their support.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Introduction

Pattern recognition can be viewed as a two-fold task, [1,2,3] namely,

- development of a decision rule based on previous knowledge (learning),
- application of the decision rule for taking decisions regarding an unknown pattern (classification).

The first task can involve one or more subtasks. For instance, it may require the design of a classifier on the basis of whatever prior knowledge there is of the feature space, or given the design, to estimate efficiently the parameters of the classifier. The latter might involve the estimation of the density function itself if very little is known about the class-conditional feature distribution, or it may necessitate the estimation of the parameters of the feature distribution, if one can assume it to have some known form. It may also involve estimating the boundaries of the classes, if even less is known about the feature space. A brief discussion regarding learning can be found in the next few sections.

All learning activities require the assistance of a set of samples from the feature space, which is called the *training set*. If the correct labels of the samples in the training set is known, the learning that takes place with the help of these samples is called *supervised learning*. Otherwise, it is termed *nonsupervised learning*. One special case of nonsupervised learning is *self-supervised learning* in which the system is equipped with a feedback mechanism so that it can learn from its past actions. In most practical situations it is either expensive or difficult to provide labels for the training samples, so that there is every possibility of having to learn with mislabeled ones. This may be due to random or systematic errors of observation or of the labeling process itself. In such situations, traditional approaches have to be modified so as to ensure that the characteristics of the method being used are not vitiated in this type of non-ideal situation.

Chapter 1

Introduction

1.1 Introduction

Pattern recognition can be viewed as a two-fold task, [1,2,3] namely,

- development of a decision rule based on previous knowledge (learning),
- application of the decision rule for taking decisions regarding an unknown pattern (classification).

The first task can involve one or more subtasks. For instance, it may require the design of a classifier on the basis of whatever prior knowledge there is of the feature space, or given the design, to estimate efficiently the parameters of the classifier. The latter might involve the estimation of the density function itself if very little is known about the class-conditional feature distribution, or it may necessitate the estimation of the parameters of the feature distribution, if one can assume it to have some known form. It may also involve estimating the boundaries of the classes, if even less is known about the feature space. A brief discussion regarding learning can be found in the next few sections.

All learning activities require the assistance of a set of samples from the feature space, which is called the *training set*. If the correct labels of the samples in the training set is known, the learning that takes place with the help of these samples is called *supervised learning*. Otherwise, it is termed *nonsupervised learning*. One special case of nonsupervised learning is *self-supervised learning* in which the system is equipped with a feedback mechanism so that it can learn from its past actions. In most practical situations it is either expensive or difficult to provide labels for the training samples, so that there is every possibility of having to learn with mislabeled ones. This may be due to random or systematic errors of observation or of the labeling process itself. In such situations, traditional approaches have to be modified so as to ensure that the characteristics of the method being used are not vitiated in this type of non-ideal situation.

Finally, chapter 6 sums up the contributions made by this thesis to the theory of recursive estimation of parameters in the field of pattern recognition, particularly, when there is a likelihood of training samples being mislabeled. It also contains suggestions for further research in this direction.

1.2 Learning

Learning has been of interest to psychologists and mathematicians for decades and more recently to computer scientists. The interest of a psychologist or a mathematician in learning is to explain or describe the manner in which animals and men learn to do a variety of skills by observing the changes in their behaviour. Such an approach is termed a *descriptive* approach. A large number of models have been developed [11,12,13] for the purpose of describing mathematically the type of learning involved here. On the other hand, in systems theory and computer science, the aim is to develop a computer program or build a machine which will 'learn' to perform certain prespecified tasks. Such an approach is called the *prescriptive* approach [14].

Learning is often associated with a goal or a performance measure. For lack of sufficient information the goal of learning may not be completely specified. In this context, learning has a dual role [15]:

1) Compensate for insufficient information by appropriate data collection and processing.

2) In that process, incrementally move towards the ultimate goal.

In systems theory and computer science, learning has been implemented in many ways :

1. The use of stochastic approximation methods [15,16,17]
2. Inductive inferential techniques [18].
3. Statistical inference techniques [19,20]
4. Heuristic programming and other Artificial Intelligence (AI) techniques [21,22]
5. Automaton models [23,24,14]
6. The application of Neural Networks [25,26,27,28,29,30,31]

Of these approaches, only the first has an immediate and direct relationship with the scope of this thesis. Among the others, the approach based on neural networks is currently generating a lot of interest among pattern recognition scientists. So a brief survey of these will be made in the later sections.

A system is required to perform the task of learning only if a priori knowledge about the system is incomplete. In general, the system can be optimized only under the assumption that its characteristics are completely known. If such knowledge is not available, one approach is to acquire the pertinent data from the actual measurements of the process as a source of information for the design or the construction of the system.

This design should approach an optimal design, and as a consequence, the performance of the system should gradually improve overall. The process of acquisition of the relevant information during the system's operation, for the purpose of improvement of the performance of the system is usually called *learning*, according to Fu [32]. Further, the problems of learning may be viewed as problems of estimation or successive approximation of the unknown quantities of a functional which is chosen by the system designer or the learning system itself to represent the process under study. The parametric and nonparametric methods of estimation studied in mathematical statistics have been used as a framework for the processes of learning in a unknown environment.

In general, these methods can be considered as special cases of successive approximations of unknown quatities. The unknown quantities may be either the parameters or the form and parameters which describe a (deterministic or probabilistic) function. However as can be seen later, both cases can be formulated as problems of successive estimations of unknown parameters. An analytic method of approximating known functions is the expansion of a 'complicated' function as a convergent infinite series of terms with simpler (or otherwise more appropriate) functional form. These expansions may be regarded as infinite approximation processes in which the error may be made arbitrarily small by taking progressively more terms into account. In general, the theory of approximations is not concerned with just continuous functions defined over an arbitrary measure space [32].

1.2.1 A basic mathematical model of learning [32]

In this section the problem of learning in an unknown stationary environment is defined as a problem of successive approximations of unknown quantities (coefficients or parameters) belonging to a preselected set of quantities which have to be estimated (learned). Let $\{\Omega_X, \mathcal{F}, p\}$ be a probability space. Ω_X is a set of elementary events (observation or feature space). $p(X)$ is a probability measure defined over Ω_X but unknown *a priori*, and $\mathcal{F}$ is a σ-algebra of subsets in Ω_X. All set functions defined over Ω_X are assumed to be real-valued, nonnegative, and integrable with respect to $p(X)$ over Ω_X.If Ω_X is a Euclidean space, the functions are also assumed to be Lebesgue-integrable over Ω_X.

Let $p(z|X,\omega\ )$ be the conditional probability density function of a random variable z for $X \in \Omega_X$ and $\omega\ \in \Omega_c$, which is not known *a priori*. Ω_c is a countable set of pattern classes or actions (or stochastic processes in general). The complement of a pattern class or action ω with respect to Ω_c is denoted by $\bar{\omega}$. It is assumed that for every $X \in \Omega_X$ and $\omega\ \in \Omega_c$,

$$\mathcal{E}[|z|\ |X,\omega\ ] = \int_{-\infty}^{\infty} |z| p(z|X,w)dz < \infty$$

$$\mathcal{E}[z^2|X,\omega\ ] = \int_{-\infty}^{\infty} z^2 p(z|X,\omega\ )dz < \infty$$

and that

$$f(X;\omega\ ) = \mathcal{E}[z|X,\omega\ ] = \int_{-\infty}^{\infty} zp(zX,\omega\ )dz$$

is real, single-valued, nonnegative, and bounded over Ω_X for $\omega\ \in \Omega_c$.

z is the performance evaluation of the classifications of $X \in \Omega_X$ into $\omega \in \Omega_c$ (or of a prediction that ω will occur, or of a decision to apply action or policy ω, after an $X \in \Omega_X$ was observed from the environment), and it is generally given by a prespecified positive-definite function

$$z = \phi(X', \omega, X).$$

X' is the observed current response of the environment due to the applied action $\omega \in \Omega_c$ following the occurrence of X; $X' \in \Omega_X$. $f(X;\omega)$ is in general the performance index function for the action, prediction, or classification $\omega \in \Omega_c$ defined over Ω_X.

The problem of learning can be stated as successively determining

$$\omega^*(X) \text{ such that} f(X;\omega^*) = \max_{\omega} \{f(X;\omega), \omega \in \Omega_c\}$$

or finding a probability measure $P(\omega|X)$ over Ω_X such that

$$P(\omega|X) = P(f(X;\omega)) = \max_{\omega} \{f(X;\omega), \omega \in \Omega_X\} \tag{1.1}$$

Here $P(\omega|X)$ should be interpreted as

1. the probability that the classification, prediction, or action ω, following immediately after the occurrence of $X \in \Omega_X$, will be correct, optimal, favorable, or 'rewarded', or

2. as the probability that the observed X belongs to the pattern class ω. If the probability $P(\omega)$, $\sum_{\omega \in \Omega_c} P(\omega) = 1$, is known, an equivalent problem is to find the conditional probability function $p(X|\omega)$ such that

$$\int_{\Omega_X} p(X|\omega)dX = 1$$

and

$$P(\omega|X) = \frac{P(\omega)p(X|\omega)}{p(X)}$$

Frequently the form of the function $p(X|\omega)$ will be known except for a certain parameter $\theta \in \Omega_\theta$. In that case the successive estimates of

$$\{p(X|\omega, \theta) : \omega \in \Omega_c, \theta(\omega) \in \Omega_\theta\}$$

are obtained by successive approximations of the parameters $\{\theta(\omega) : \theta(\omega) \in \Omega_\theta, \omega \in \Omega_c\}$. The basic problem of learning in an unknown stationary environment is then reduced to that of successively establishing $f(X;\omega)$ or $p(X|\omega)$ on the basis of observations $\{z_n(X_n;\omega) : n = 1, 2, \ldots\}$ which are distributed according to $\{p(z|X_n, \omega) : n = 1, 2, \ldots\}$ where $\{X_n : n = 1, 2, \ldots\}$ is a sequence of elementary events identically distributed over Ω_X by

$$p(X) = \sum_{\omega \in \Omega_c} P(\omega)p(X|\omega).$$

The convergence of estimation assures us that the estimated value of the unknown quantity will approach its true value. Consequently, the design or decision based

the estimated information will eventually approach a desired optimal design or decision rule.

The approaches of using Markov models or reinforcement learning algorithms are actually based on the equation 1.1, where Ω_X is a countable set. A normalized measure of optimality is defined for the equation 1.1 by

$$P(\omega\,|X) = \frac{f(X;\omega\,)g(f(X;\omega\,))}{\sum_{\omega\,'\in\Omega_c} f(X;\omega\,')g(f(X;\omega\,'))}$$

where $g(\cdot)$ is a positive nondecreasing real-valued function defined on the real line. In those reinforcement models where a reward or 'positive reinforcement' is associated with $z = +1$ in the case of a correct recognition, prediction, or action, and a 'penalty' or 'negative reinforcement' with $z = 0$ in the case of misrecognition of X, or wrong prediction or action,

$$f(X;\omega\,) = \mathcal{E}[z|X,\omega\,] = P\{z = +1|X,\omega\,\}$$

and

$$P(\omega\,|X) = P[z = +1|X,\omega\,] = f(X;\omega\,)$$

such that for every X

$$\sum_{\omega\,\in\Omega_c} P(\omega\,|X) = 1.$$

The learning automata using "linear tactic" models can be described by the above if a simple normalization of their states is performed at each step when a reward or penalty is obtained.

Point estimation of $\theta \in \Omega_\theta$ in $p(X|\omega\,,\theta)$ can be performed by successive applications of Bayes' formula; such a process is usually termed *Bayesian learning.*

1.3 Learning in a pattern recognition system

All the numerous classifiers known to pattern recognition scientists can be implemented with complete *a priori* knowledge relevant to pattern classes, namely, weighting coefficients in linear discriminant classifer, reference vectors in minimum distance classifier and $P(\mathcal{C}_i)$ and $p(\mathbf{X}|\mathcal{C}_i)$ in Bayes' classifier etc. It should be understood that in practice, an infinite number of samples of classes are not available. We have, instead, a finite and usually small number of samples so that information required for optimal design of feature extractor or classifiers is often partially known. Under such circumstances, we must assume at best, that these samples are representative of those which would be obtained by examining a much larger sample size. Such a set of typical patterns is called a *training set.* If this requirement is satisfied — and we must usually satisfy this requirement through engineering judgement exercised in the selection of samples — the classifier can be designed to have the capability of learning the best values of the weights/statistical information from the training patterns to result in nearly the minimum number of misclassification [32].

Any method of selection and/or placement of hypersurfaces employing a training set can then be called a training method and a classifier whose hypersurfaces are

Any method of selection and/or placement of hypersurfaces employing a training set can then be called a training method and a classifier whose hypersurfaces are adjustable can be called a *trainable pattern classifier*. The objective of all our efforts is to design optimum trainable classifying systems, which comes under the popular heading of *Learning Machines* [33,34,35,36,16,20,32,17].

Learning is thus a task of constructing the regions or templates in the N-dimensional space in which labeled samples of the classes are contained. By observing the patterns with known classification, a linear discriminant classifier, for example, can automatically adjust the weighting coefficients associated with its discriminant function. The performance of the classifier is supposed to improve up to a point as the number of training patterns is increased. Under the assumption that the patterns from different classes are linearly separable, it is also possible to develop several algorithms referred to as 'error-correction' training procedures to find the linear hyperplanes which properly separate the data and to have the property of converging to the solution which linearly separates the prototypes into their correct classifications if indeed the data is so separable.

The two stages of pattern recognition, namely, deriving the decision rule (learning), and using it to recognise a pattern, can be performed in two ways —

(a) learning before recognition and

(b) learning and recognition concurrently.

In the first method all the labeled pattern samples are collected and the best decision rule based on those samples is derived. This fixed decision rule is then applied without change to classify unlabeled patterns. Whereas, the decision rule in the second method is adaptive and is updated according to output decision. If the learned information gradually approaches the true information, then the decisions based on the learned information will eventually approach the optimal decision as if all the information required is known. Therefore, during the system's operation, the performance of the system is gradually improved. A learning pattern recognition system too, can be termed as 'supervised' or 'nonsupervised' depending on whether the correct classification of the input patterns observed are known or not. Similarly, a classifier that learns from its own past 'experiences' is said to be self-supervised.

1.3.1 Parametric and non-parametric learning

In the statistical classification approach, if the unknown information is the parameter values of a known distribution function $p(\mathbf{X} \mid C_i)$, the *parametric learning* technique can be applied. If the functional form of this function is known, then the well-developed theory of statistical point estimation can easily be used to good advantage. For instance, if $\Phi_1, \Phi_2, \ldots, \Phi_m$ are the m training subsets of patterns corresponding to the m classes, then with the knowledge that $p(\mathbf{X}|C_i), i = 1, 2, \ldots, m$ are Gaussian, the estimates of the parameters μ_i and Σ_i are defined by the

following sample statistics :

$$\langle \mathbf{X} \rangle_j = \frac{1}{M_j} \sum_{\mathbf{X} \in \Phi_j} \mathbf{X}$$

$$\langle \mathbf{S} \rangle_j = \frac{1}{M_j} \sum_{\mathbf{X} \in \Phi_j} (\mathbf{X} - \langle \mathbf{X} \rangle_j)(\mathbf{X} - \langle \mathbf{X} \rangle_j)^T,$$

where M_j is the number of patterns in the training subset Φ_j, and $\langle \mathbf{X} \rangle_j$ and $\langle \mathbf{S} \rangle_j$ denote the sample mean (or, centre of gravity) and sample covariance matrix respectively of the samples in Φ_j.

Nonparametric learning

If the class-conditional densities are not easily described in terms of a small set of parameters, and if the optimum decision surface is known to be highly nonlinear, one should consider nonparametric training techniques for estimating these densities. The theory of nonparametric estimation in statistics is reasonably well-developed, and can be applied to good advantage. It is possible to get nonparametric point estimates of class parameters as well as estimates of the unknown density functions. Nonparametric estimates have the added advantage that they are robust, in general.

Of the two types of nonparametric estimates mentioned above, the problem of estimation of the density functions is the more important. Basically, it is a problem of estimation of a function of several variables. It can be used even when there is no *a priori* knowledge of the density. However, when sufficient knowledge is available, it is more advisable to fit a suitable density function to the data. Nonparametric training techniques exploit the fact that for many practical situations the class densities are smooth functions over the sample space. (By a smooth function we mean that changes in values of the function are relatively small in any small neighborhood of a point in the function's domain and, if the domain is a continuum, that these changes become infinitesimal as the size of the neighborhood approaches zero.) The assumption of smoothness in the class densities leads to the replacement of each observed feature vector by a positive, single-peaked, piecewise continuous function that contributes linearly to an estimate of the class density. We refer to this replacement as *smoothing*. The smoothing techniques commonly used and to be described briefly in the paragraphs to follow are :

(a) *Parzen estimates*: The Parzen estimate $\hat{p}_n(x)$of a univariate density $p(x)$, based on a sample of size n, is

$$\hat{p}_n(x) = \frac{1}{n} \sum_{i=1}^{n} \frac{1}{h} k((x - x_i)/h),$$

where h is a positive number, suitably chosen, possibly a fuction of n which tends to 0 as n tends to ∞, and

$$k(c) = \begin{cases} 1/2 & \text{for } c \leq 1 \\ 0 & \text{otherwise} \end{cases}$$

and the term $\frac{1}{h}k(\cdot/h)$ is called the *kernel* of the estimate. If $k(\cdot)$ and h satisfy certain conditions [19], it can be shown that the estimate is asymptotically unbiased and consistent. The estimate can easily be generalized to the case of multivariate densities as follows:

$$\hat{p}_n(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{h^n}k((\mathbf{x}-\mathbf{x}_i)/h),$$

or, more generally,

$$\hat{p}_n(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{\prod_{j=1}^{n}h_j}k((x_1-x_{1i})/h_1,(x_2-x_{2i})/h_2,\ldots,(x_n-x_{ni})/h_n).$$

These estimates too, are asymptotically unbiased and consistent.

(b) *k-nearest neighbour estimates* [37,34]: For a given value of k, the k-nearest neighbour estimate of a density $p(\mathbf{X})$ is

$$\hat{p}_n(\mathbf{x}) = \frac{k-1}{n}\frac{1}{A(k,n,\mathbf{X})},$$

where $A(k,n,\mathbf{X})$ is the volume of the smallest set containing the k nearest neighbours of $\mathbf{X}$ in the training set, and is a random variable, depending on the selected set of n samples. It can be proved that these estimates too, are asymptotically unbiased and consistent.

(c) *Histogram or bar graph estimates* [19,34]: By treating both k and A as defined just above, as variables it is possible to obtain other estimates of the density function, which are called histogram estimates. The sample space is partitioned into mutually disjoint cells, and the density function is approximated by the number of samples which fall in each cell. The cell size can be either fixed or variable. By using variable cell sizes it is possible to reduce the number of cells, since for a fixed cell size, the number of cells can be prohibitively large, being M^n for n variables with M cells per variable. The estimate so obtained is

$$\hat{p}_n(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n}\psi_n(\mathbf{X}-\mathbf{X}_i,V_n),$$

where

$$\psi_n(\mathbf{X},V_n) = \begin{cases} \frac{1}{V_n} & \text{if } \mathbf{X} \text{ and } \mathbf{X}_i \in C_{nk} \text{ for some } k \\ 0 & \text{otherwise} \end{cases}$$

is the *window function* and C_{nk} is the kth cell at the nth iteration and V_n is its hypervolume, being equal for all cells. These estimates can be shown to converge in probability to $P(\mathbf{X})$.

(d) *Estimates based on basis functions*: This approach estimates a density function by finding an expansion for it in a set of basis functions $\phi_i(X)$ as

$$p(X) = \sum_{i=1}^{\infty}c_i\phi_i(X).$$

If the basis functions are orthogonal with respect to the kernel $k(X)$, that is, if

$$\int k(X)\phi_i(X)\phi_i^*(X)dX = \lambda_i \delta ij,$$

where $\phi_i^*(X)$ is the complex conjugate of ϕ_i and δ_{ij} is the Kronecker delta, then the coefficients are given by

$$.\lambda_i c_i = \int k(X)\phi_i(X)\phi_i^*(X)dX.$$

In the one-dimensional case, a number of such basis functions are available, such as Fourier series, Legendre, Jacobi, Hermite and Leguerre polynomials.

(e) *Estimates obtained by potential functions*: This is a special case of the last method, in which successive estimates of the coefficients c_i are obtained by using a special type of kernel called a potential function $k(\cdot,\cdot)$ which satisfies the conditions

$$k(Y,X) = k(X,Y)$$

and

$$k(Y,X) = \sum_{i=1}^{\infty} \lambda_i{}^2 \phi_i(Y)\phi_i(X).$$

The successive estimates obtained are

$$\hat{p}_{n+1}(X) = \hat{p}_n(X) - \gamma_n k(X_n, X)$$

where

$$\gamma_n = 2a_n\{\hat{p}_n(X_n) - f(X_n)\}$$

and

$$k(X_n, X) = \Phi(X_n)^T \Phi(X),$$

where

$$\Phi = [\phi_1\ \phi_2\ \cdots\ \phi_\infty]^T,$$

$f(\cdot)$ is an observable random variable whose expectation is $p(\cdot)$, and $\{a_n\}$ is a sequence of positive numbers satisfying

$$a_n > 0,$$

$$\lim_{n\to\infty} a_n = 0,$$

$$\sum_{n=1}^{\infty} a_n = \infty$$

and

$$\sum_{n=1}^{\infty} a_n{}^2 < \infty.$$

These estimates can be shown to converge to $p(X)$ in the mean square as well as with probability 1.

Stochastic approximation too, is a nonparametric procedure, but on account of its great proximity to the subject matter of this thesis, is covered in a separate section.

Another aspect of nonparametric learning in pattern recognition is the estimation of coefficients of discriminant functions [37,19,38], linear discriminant functions, in particular. Even if the discriminant is not linear it is possible to transform the variables so that the resulting discriminant is linear in the transformed space. It is possible, therefore, to express a discriminant function $g(\mathbf{x})$ as

$$g(\mathbf{x}) = \mathbf{w}^T\mathbf{x},$$

where $\mathbf{w}$ is the vector of coefficients and $\mathbf{x}$ is the (transformed) feature vector. It is possible to use *linear programming techniques* as well for estimating $\mathbf{w}$ [37].

Two types of situations are generally encountered, one in which the classes are *linearly separable* and one in which they are not. Incidentally, two classes are said to be linearly separable if it is possible to find a linear hyperplane which separates the two. If the classes are linearly separable, it is possible to use a class of nonparametric learning procedures, called *gradient descent procedures* for estimating the coefficients recursively. It is an important class of procedures and is described briefly below, together with some of its more important special cases.

The gradient descent technique

This technique can be described very accurately with the help of an example from everyday life, which is given by Sklansky and Wassel in their book [34]. Let us visualize a man standing on a hillside in a very dense fog. He wishes to find his home, which is located at the point of lowest altitude within a very large crater. If he begins his search within the crater, if the terrain is smooth, and if there are no dips or smaller craters along his path, he will find his home by always traveling against the direction of maximum increasing slope. In doing this, he would be applying the gradient descent technique.

Conceptually, then, the gradient descent technique is simple. A loss function $J(\mathbf{w})$ is first determined, where $\mathbf{w}$ is a controllable parameter vector. (In the example of the man on the hillside, $\mathbf{w}$ is a position vector.) One seeks the value or values of $\mathbf{w}$ where $J(\mathbf{w})$ is a minimum. In the example, the loss function $J(\mathbf{w})$ is the altitude of the terrain as a function of $\mathbf{w}$. Let $\mathbf{w}(t)$ denote $\mathbf{w}$ as a function of time t. An initial vector $\mathbf{w}(0)$ is arbitrarily selected. Motion of the parameter vector commences in parameter space from this point. The motion is always in a direction which is exactly opposite to the gradient of the loss function (i.e., the direction of $-\nabla J(\mathbf{w})$ which is the negative of the direction of maximum increasing $J(\mathbf{w})$ at location $\mathbf{w}$). If a loss function has been chosen so that the descent motion does not become entrapped at a local minimum and if a finite global minimum exists, the gradient descent procedure will find a parameter vector $\mathbf{w}$ such that $J(\mathbf{w})$ is minimized.

For recursive techniques some modification of this technique is made since discrete corrections to $\mathbf{w}$ require that the descent motion be an incremental approximation

to the continuous gradient descent path. In these modified procedures, we converge to a minimum of a function $J(\mathbf{w})$ by making an initial guess $\mathbf{w}(0)$, finding the gradient (multidimensional derivative) of $J(\mathbf{w})$ at $\mathbf{w} = \mathbf{w}(0)$, and making a second guess $\mathbf{w}(1)$ by adding to $\mathbf{w}(0)$ a vector having the direction of the negative of the gradient. Subsequent $\mathbf{w}(k)$'s are computed in a similar manner. Under proper constraints on $J(\mathbf{w})$ and $\mathbf{w}(0)$, $\{v(k)\}$ will converge to a vector $\mathbf{w}^*$ where $J(\mathbf{w}^*)$ is a local minimum.

Gradient descent procedures often use the following recursive form :

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \rho_k[-\nabla J(\mathbf{w}(k))],$$

where ρ_k is a positive number that may be constant or may depend on k.

In pattern recognition training procedures, a complete description of the class probability densities is generally unavailable. Therefore, the actual gradient at a point cannot be calculated; instead, a statistical estimate of the gradient using a finite sample must be used. When this estimate is used, the training procedure becomes stochastic approximation rather than gradient descent.

For different forms of the loss function $J(\cdot)$, different learning procedures can be obtained, like the perceptron algorithm, the fixed-increment procedure, the variable increment procedure and the relaxation procedures [37]. If the loss function has the form

$$J(\mathbf{w}) = \sum_{i=1}^{n} (\mathbf{w}^T \mathbf{x}_i - b_i)$$

where the b_i are some prespecified positive constants, then it is possible to get minimum-squared error estimates by the gradient search technique. A variant of this technique is the *Widrow-Hoff procedure*. Another is the Ho-Kashyap procedure which uses the loss function

$$J(\mathbf{w}, \mathbf{b}) = \sum_{i=1}^{n} (\mathbf{w}^T \mathbf{x}_i - b_i)^2$$

and minimises J with respect to both $\mathbf{w}$ and $\mathbf{b}$, and hence obtains estimates for both. This is quite useful as $\mathbf{b}$ is generally not known beforehand.

1.3.2 Nonsupervised learning

As noted earlier, if there is no teacher for providing the correct label for each observation of the system, then the situation calls for a special type of learning called nonsupervised learning. Very often there is no information as to the number of classes. In such cases, one speaks in terms of regions or clusters that can be separated with the help of a decision rule, and not about actual classes. Therefore, nonsupervised learning is discussed in terms of *clustering*. The theory of clustering has seen quite a bit of research, and any number of algorithms are available [39, 40, 41].

1.3.3 Bayesian learning

Here the Bayesian decision-theoretic approach is applied to do the learning, both supervised and nonsupervised [42]. The basic premise of the Bayes approach to decision-making is that the decision d itself is a random variable defined over the decision space Ω_d. A distribution for the decision, known as the prior distribution, is assumed. A Bayesian decision rule is one which minimizes the *posterior risk*, which is nothing but the expected value of the risk function with respect to the *posterior density* $p(d|\mathbf{X})$. Since the estimates of parameters are random variables, we will see in this section how the density function of the estimate can be calculated by a successive process. Both the supervised and unsupervised techniques using Bayes' theorem are treated separately.

Supervised learning

Supervised estimation schemes based on Bayesian learning that can be used to obtain successive estimates of an unknown parameter θ for each class C_i, of a feature distribution $p(X|C_i)$ whose functional form is fully known.

Let the *a priori* density function for the unknown parameter θ be $p_0(\theta\ )$ which reflects the initial knowledge about θ (an N-dimensional vector). Let

$$\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_n$$

be a sequence of independent identically distributed feature vectors observed from the same pattern class C_i. Then according to Bayes' theorem, *a posteriori* density function of θ given the first observation $\mathbf{X}_1$ is

$$p(\theta\ |\mathbf{X}_1) = \frac{p(\mathbf{X}_1|\theta\ )p_0(\theta\ )}{p(\mathbf{X}_1)}.$$

After $\mathbf{X}_1$ and $\mathbf{X}_2$ are observed, the *a posteriori* density function of θ is

$$p(\theta\ |\mathbf{X}_1.\mathbf{X}_2) = \frac{p(\mathbf{X}_2|\mathbf{X}_1, \theta\ )p(\theta\ |\mathbf{X}_1)}{p(\mathbf{X}_2|\mathbf{X}_1)}.$$

After the nth observation is observed, it becomes

$$p(\theta\ |\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_n) = \frac{p(\mathbf{X}_n|\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_{n-1}, \theta\ )p(\theta\ |\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_{n-1})}{p(\mathbf{X}_n|\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_{n-1})}.$$

With the knowledge of $p(\theta\ |\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_n)$ one can compute the required probability density function

$$p(\mathbf{X}_{n+1}|\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_n, C_i) =$$

$$\int p(\mathbf{X}_{n+1}|\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_n, C_i, \theta\ )p(\theta\ |\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_n, C_i),$$

where $p(\mathbf{X}_{n+1}|\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_n, \theta\ )$ is known. The *a posteriori* density function on the average, becomes more concentrated and the estimate converges to the true value of the parameter so long as the true value is not excluded by the *a posteriori* density function of the parameter θ .

Nonsupervised learning

In nonsupervised learning, the training samples (as their correct classifications are not known) are considered as coming from the mixture distribution having the probability distributions of all the classes as component distributions. The problem of learning is then reduced to a process of successive estimation of some unknown parameters in either a mixture distribution of all possible pattern classes or a known decision boundary.

The mixture distribution is defined as

$$p(\mathbf{X}) = \sum_{i=1}^{W} p(\mathbf{X}|Z_i^{(n)})P(Z_i^{(n)}),$$

where

$p(\mathbf{X}|Z_i^{(n)})$ denotes the ith-partition conditional distribution,

$P(Z_i^{(n)})$ the mixing parameter for ith-partition $Z_i^{(n)}$

and

$$W(= m^n, m = \text{number of class distributions})$$

is the number of ways $Z_1^{(n)}, Z_2^{(n)}, \ldots, Z_W^{(n)}$ in which the set of training observations $\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_n$ can be partitioned constituting an overall mixture distribution.

If $p(\mathbf{X}|\Theta, \mathcal{P})$ represents the parameter-conditional mixture distribution where $\Theta = \{\theta_1, \theta_2, \ldots, \theta_W\}$ and $\mathcal{P} = \{P(Z_1^{(n)}), P(Z_2^{(n)}), \ldots, P(Z_W^{(n)})\}$ are the two sets of parameters and $p(\mathbf{X}|\theta_i, Z_i^{(n)})$ the ith parameter-conditional distribution, then in terms of the set of parameters the above equation becomes

$$p(\mathbf{X}|\Theta, \mathcal{P}) = \sum_{i=1}^{W} p(\mathbf{X}|\theta_i, Z_i^{(n)})P(Z_i^{(n)}).$$

The problem of nonsupervised learning is therefore reduced to that of finding a unique solution for Θ and $\mathcal{P}$, given $p(\mathbf{X}|\Theta, \mathcal{P})$.

Let us now assume that there are two pattern classes C_1 and C_2 having the respective known form of the probability density functions $p(\mathbf{X}|C_1)$ and $p(\mathbf{X}|C_2)$, and the parameter θ of the mixture distribution is unknown. Then the *a posteriori* density for obtaining the (nonsupervised) Bayes estimate of the parameter θ is,

$$p(\theta|\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_n) = \sum_{i=1}^{W} p(\theta|\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_n, Z_i^{(n)})p(Z_i^{(n)}|\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_{n-1}),$$

where $W = 2^n$. The problem is therefore reduced to that of supervised learning for each of the 2^n partitions.

1.4 Learning using stochastic approximation

Stochastic approximation is a recursive nonparametric gradient descent-type technique that has been developed as an optimization technique for random environments. This approximation can be used for successive estimation of an unknown parameter, when due to the stochastic nature of the problem, the measurements are expected to have certain errors. The technique guarantees the convergence of the algorithm even when the observation vectors are not linearly separable. Details about stochastic approximation along with the several applications such as in communication theory, control theory and pattern recognition, are available in texts [43,15,44]. The present section relates only to some of the learning methods in pattern recognition problems using stochastic approximation. In this context, it is worthwhile to consider first a simple example which leads to a basic approach to successive estimation, and to the notion of stochastic approximation. Let there be n observation vectors $\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_n$ and a successive estimate of the mean vector μ is required from these observations. The non-successive estimate $\bar{\mathbf{X}}_n$ of the mean vector based on these observations is given by

$$\bar{\mathbf{X}}_n = \frac{1}{n}\sum_{i=1}^{n}\mathbf{X}_i.$$

This can be rewritten as

$$\bar{\mathbf{X}}_n = \frac{1}{n}\sum_{i=1}^{n-1}\mathbf{X}_i + \frac{1}{n}\mathbf{X}_n$$

or

$$\begin{aligned}\bar{\mathbf{X}}_n &= \frac{n-1}{n}\frac{1}{n-1}\sum_{i=1}^{n-1}\mathbf{X}_i + \frac{1}{n}\mathbf{X}_n \\ &= \frac{n-1}{n}\bar{\mathbf{X}}_{n-1} + \frac{1}{n}\mathbf{X}_n \\ &= \bar{\mathbf{X}}_{n-1} + \frac{1}{n}(\mathbf{X}_n - \bar{\mathbf{X}}_{n-1})\end{aligned}$$

Therefore, if we store n and $\bar{\mathbf{X}}_{n-1}$, the mean vector estimated from $(n-1)$ samples, we can compute $\bar{\mathbf{X}}_n$ with a new incoming nth observation $\mathbf{X}_n$ using the last equation. It also shows that as n increases, the effect of the new sample $\mathbf{X}_n$ on the expected vector decreases as follows :

$$\mathbf{X}_1, \frac{1}{2}\mathbf{X}_2, \frac{1}{3}\mathbf{X}_3, \ldots, \frac{1}{i}\mathbf{X}_i, \ldots, \frac{1}{n}\mathbf{X}_n$$

The sequence $1, \frac{1}{2}, \frac{1}{3}, \ldots, \frac{1}{i}, \ldots, \frac{1}{n}$ is known as a 'harmonic sequence'. The above findings therefore suggest that if we have an expression for non-successive estimate of a parameter from n samples, the expresion for its successive estimate may be obtained by separating the estimate in two parts, one of which corresponds to the estimate obtained from $(n-1)$ samples and the other is the contribution of the

nth sample. The effect of the nth sample can also be made smaller by using a coefficient which is a decreasing function of n. This is the precisely the principle underlying the nonparametric technique of stochastic approximation.

Stochastic approximation is a very useful technique for recursive parameter estimation in pattern recognition, and convergence is guaranteed under fairly general conditions, although it is usually difficult to get an idea of the rate of convergence.

The earliest application of this technique was to find a root of a regression function, and the procedure so obtained is the well- known Robbins-Monro algorithm [45]. If θ and $\mathbf{z}$ be two random variables with some correlation then the problem is to find a root of the regression function $f(\theta)$ which is given by

$$f(\theta) = \mathcal{E}(\mathbf{z}|\theta).$$

The Robbins-Monro algorithm for this is

$$\theta_{n+1} = \theta_n - a_n \mathbf{z}_n,$$

where $\mathbf{z}_n$ is the nth observation on $\mathbf{z}$. Robbins and Monro proved that the algorithm converges in mean square provided the sequence $\{a_n\}$ satisfies the following conditions:

$$\lim_{n\to\infty} a_n = 0,$$

$$\sum_{n=1}^{\infty} a_n = \infty,$$

$$\sum_{n=1}^{\infty} {a_n}^2 < \infty.$$

Later, Blum [46] established that the algorithm also converges with probability 1. Dvoretzky [47] provided a generalised form of the convergence proofs of Robbins-Monro and Blum and showed that the two modes of convergence hold for any stochastic convergence procedure that satisfies the conditions of his theorem.

Kiefer and Wolfowitz [48], taking the Robbins- Monro,method as their point departure, considered the problem of obtaining the extremum of a regression function, and arrived at a stochastic approximation procedure as the solution, which bears their names. Since then a lot of research has taken place in this field [43,49,44], and it continues to be a popular estimation technique.

1.5 Learning with Networks of Neuron-like elements

During the mid 1950's and early 1960's a class of machines, first proposed by Rosenblatt [50] seemed to offer what many researchers thought was a natural and powerful model of machine learning. These machines were christened perceptrons, and are the precursors of the neural networks that were formulated later. Minsky and Papert, in one of the pioneering books in this area [51], defined a perceptron as a device capable of computing all predicates which are linear in some given set of partial predicates.

The basic model of a perceptron capable of classifying a pattern into one of two classes, as described, for instance, by Tou and Gonzalez [38], envisages a machine consisting of an array S of sensory units which are randomly connected to a second array A of associative units. Each of these units produces an output only if enough of the sensory units that are connected to it are activated. These sensory units may be looked upon as the means by which the machine receives stimuli from its external environment, that is, its measurement devices and the associative units as the first stage or input to the machine.

The response of the machine is proportional to the weighted sum of the associative array responses, that is, if x_i is the response of the ith associative unit and w_i the corresponding weight, the response is

$$R = \sum_{i=1}^{n+1} w_i x_i.$$

If $R > O$, then the pattern observed by the sensory units belongs to class C_1; otherwise, it belongs to class C_2. This can easily be extended to the m-class case, there being m responses $R_1, R_2, \ldots, R_m$ in this case. The pattern is assigned to the class C_i if $R_i > R_j$ for all $j \neq i$. The basic model can easily be extended to nonlinear decision functions by inserting the appropriate nonlinear preprocessor between the A and R arrays.

The training algorithm for the perceptron described above is a simple scheme for the iterative determination of the weight vector $\mathbf{w}$. The scheme, frequently called the perceptron algorithm is as follows :

Given two training sets belonging to pattern classes C_1 and C_2 respectively, let $\mathbf{w}(1)$ be the initial weight vector, which may be arbitrarily chosen. Then, at the kth training step

$$\text{if} \quad \mathbf{x}(k) \in C_1 \quad \text{and} \quad \mathbf{w}^T(k)\mathbf{x}(k) \leq 0$$
$$\text{replace} \quad \mathbf{w}(k) \quad \text{by} \ \mathbf{w}(k+1) = \mathbf{w}(k) + c\mathbf{x}(k)$$

and

$$\text{if} \quad \mathbf{x}(k) \in C_2 \quad \text{and} \quad \mathbf{w}^T(k)\mathbf{x}(k) \geq 0$$
$$\text{replace} \quad \mathbf{w}(k) \quad \text{by} \ \mathbf{w}(k+1) = \mathbf{w}(k) - c\mathbf{x}(k)$$

Otherwise, leave $w(k)$ unchanged, that is, $\mathbf{w}(k+1) = \mathbf{w}(k)$.

In other words, the algorithm makes a change in $\mathbf{w}$ if and only if the pattern being considered at the kth step is misclassified by the weight vector at this step. The correction increment c must be positive. The algorithm is, therefore, obviously a reward-and-punishment procedure. The 'reward' for taking a correct decision is to leave the weight vector unchanged, the 'punishment' for not taking a correct decision being either to increase it or to decrease it, depending upon an auxiliary condition.

It can be shown that if the pattern classes are linearly separable, then the perceptron algorithm converges, that is, yields a solution weight vector in a finite number of steps.

Several variations of the perceptron algorithm can be formulated, depending on how the value of the correction increment c is selected. Among the commonly-used training algorithms are the *fixed-increment* algorithm, the *absolute- correction* algorithm and the *fractional-correction* algorithm. In the first, c is a constant greater than zero. In the second, c is chosen to be just large enough to guarantee that the pattern is correctly classified after a weight adjustment. That is, if $\mathbf{w}^T(k)\mathbf{x}(k) \leq 0$, c is chosen so that

$$\mathbf{w}^T(k+1)\mathbf{x}(k) = \mathbf{w}(k) + c\mathbf{x}(k)]^T\mathbf{x}(k) > 0.$$

One way is to choose c as the smallest integer greater than

$$|\mathbf{w}^T(k)\mathbf{x}(k)|/\mathbf{x}^T(k)\mathbf{x}(k).$$

In the third type of algorithm, c is so chosen as to make $|\mathbf{w}^T(k)\mathbf{x}(k) - \mathbf{w}^T(k+1)\mathbf{x}(k)|$ a certain positive fraction λ of $|\mathbf{w}^T(k)\mathbf{x}(k)|$, that is,

$$|\mathbf{w}^T(k)\mathbf{x}(k) - \mathbf{w}^T(k+1)\mathbf{x}(k)| = \lambda|\mathbf{w}^T(k)\mathbf{x}(k)|.$$

Substituting $\mathbf{w}(k+1) = \mathbf{w}(k) + c\mathbf{x}(k)$ in this yields

$$c = \lambda\frac{|\mathbf{w}^T(k)\mathbf{x}(k)|}{\mathbf{x}^T(k)\mathbf{x}(k)}.$$

Clearly, the initial weight vector must be different from 0. If $\lambda > 1$, a pattern is correctly classified after each weight adjustment, and if $0 < \lambda < 2$, this algorithm can be shown to converge.

The *Adalines* (*ada*ptive *line*ar device*s*) of Widrow [52] are similar to perceptrons in the sense that they involve trainable threshold logic units. The output is linear, which can be converted to digital outputs by discriminator elements. Actually, adaline represents the simplest form of the perceptron that consists of a single threshold element.

Perceptrons as described above are actually the first of the networks of neuron-like elements (or, neural networks) that were considered for solving problems like pattern recognition, and so on. These perceptrons are now called single-layer perceptrons (SLPs) in the sense that the notion has since been generalized to that of multi-layer perceptrons (MLPs), which are nothing but feed-forward nets with one or more layers of sensory nodes between the input nodes and the output nodes [31]. These additional layers contain 'hidden' units or nodes that are not directly connected to both the input and output nodes. MLPs overcome many of the shortcomings of SLPs but were generally not used in the past because effective training algorithms were not available. This situation has changed recently with the development of new training algorithms [26]. Although it cannot be proved that these algorithms converge as with SLPs, they have been shown to be successful for many problems of interest. The capabilities of MLPs stem from the nonlinearities used within nodes. It has also been shown that no more than three layers are required in perceptron-like feed-forward nets because a three- layer net can generate arbitrarily complex decision regions.

One of the algorithms that can be used to train MLPs with multiple output nodes and sigmoidal nonlinearities is the *back- propagation* training algorithm, which is really an iterative gradient search algorithm designed to minimize the mean square error between the actual output (of the perceptron) and the desired output. It gives in general, a good performance, which is surprising considering that it is a gradient search technique that may find a local minimum in the objective function rather than a global minimum.

Of the commonly-known neural networks, the most important ones that can be used as classifiers, apart from single layer perceptrons and multi-layer perceptrons, are [31]:

- the Hopfield nets
- the Hamming nets
- the Carpenter/Grossberg classifiers
- the Kohonen self-organizing feature maps.

The Hopfield and the Hamming nets can be trained with supervision, but are generally used with fixed weights. The Hamming net is a neural net implementation of the optimum classifier for binary patterns corrupted by random noise. The Carpenter/Grossberg classifier and Kohonen's feature map do unsupervised learning, the former by the leader algorithm and the latter by the K-means algorithm.

In general, neural networks are composed of many nonlinear computational elements operating in parallel and arranged in patterns reminiscent of biological neural nets. The computational elements or nodes are connected via weights that are typically adapted during use to improve performance. The ability to adapt and continue learning is not only a highly desirable characteristic as far as practical pattern recognition is concerned, it also provides a degree of robustness by compensating for minor variabilities in characteristics of processing elements. Neural net classifiers are also non- parametric and make weaker assumptions regarding the underlying distributions than traditional statistical classifiers. As such they can be expected to be more robust in non-Gaussian situations.

An unsupervised learning paradigm that has attracted a lot of attention recently is *competitive learning* [25,29]. It has been found that when applied to parallel networks of neuron–like elememts, many potentially useful learning tasks can be accomplished. For instance, it seems to provide a way to discover the salient, general features that can be used to classify a set of patterns [29].

Chapter 2

The GGA:A generalised learning algorithm based on guard zones

2.1 Introduction

An adaptive pattern recognition system can be viewed as a learning machine in which the decision of the system gradually approaches the optimal decision by acquiring necessary information from observed patterns. System performance is improved as a result [32]. In a supervised system, the machine requires an extra source of knowledge, usually of a higher order, for correcting the decision taken by a classifier. When an extra source of knowledge on which a supervisory programme could be based is not readily available, the performance of the system becomes highly unpredictable. In particular, if a system is capable of utilizing its past experiences and behaviour while learning, is called a self-supervised system. Self-supervised learning is a special case of non-supervised learning. The proposed algorithm GGA is capable of self-supervised learning, if called upon to do so.

As already mentioned in chapter 1 (section 1.2), Bayesian estimation methods and stochastic approximation [42,15,44,49] are some of the most widely used tools for recursive learning of class parameters. Within the class of stochastic approximation-type algorithms, there exists a subclass of algorithms that essentially aim to correct for the presence of outlier-type or non-representative or 'doubtful' training samples by attempting to remove them from the training set. In this context, two algorithms can be singled out, the first being the self- supervised learning system based on the concept of a *guard zone*, mooted by Pal et al. [53]. Here, to restrict the updating of estimates of parameters (feature means and variances) by means of 'doubtful' samples, a *guard zone* was defined for each class in such a way that a training sample was used for updating only if it fell within the guard zone. The guard zone is so constructed that the probability of misclassification of the input patterns falling within it, given that it is constructed around the central tendency of the feature distribution in a class, is substantially low. The second is an algorithm presented by Chien [54] as a solution to the problem of identifying 'spurious', that is, possibly non-representative training samples, for the case when feature means are to be learned. A threshold is defined such that if the 'distance' of the current training sample from the preceding estimate

of the mean (the same 'distance' is used for defining a guard zone) exceeds it, the training sample is rejected. This thesis generalises the notion of such algorithms into the so-called Generalised Guard Zone Algorithm (GGA) and examines and compares a few of its properties *vis-à-vis* the usual stochastic approximation algorithm (to be referred to as the non-GGA). In this chapter, a formal definition of the GGA as well as the non-GGA is given in section 2.2. Section 2.3 shows how the GGA specialises to the algorithms of Pal et al. [53]and Chien [54]. Some intuitive considerations regarding such algorithms are discussed in section 2.4. Also,as such algorithms basically aim to detect outliers and reject them from the parameter-updating procedure, they can be looked upon as a robust estimation procedures [55,56,57,58,59]. This aspect is discussed in section 2.5.

2.2 The Generalized Guard Zone Algorithm

Let

$$\mathbf{X} = [X_1, X_2, \ldots, X_N]', \ \mathbf{X} \in \mathcal{R}^N$$

be an N-dimensional feature vector defined over a pattern class $\mathcal{C}$. Let the following assumptions hold true:

(**A1**) The distribution of $\mathbf{X}$ over $\mathcal{C}$ is continuous.

(**A2**) This distribution depends on a q-dimensional parameter vector θ , some or all elements of which need to be learned.

(**A3**) The distribution of $\mathbf{X}$ over $\mathcal{C}$ is such that $\mathcal{E}(\mathbf{X})$ exists and is equal to μ .

(**A4**) The dispersion matrix of $\mathbf{X}$, namely,

$$Disp(\mathbf{X}) = \mathbf{\Sigma} = ((\sigma_{ij}))$$

exists.

A guard zone is formally defined as follows :

Definition 2.1 *Let $\mathcal{S}$ be a metric space and δ a metric defined on it. Then for any point $\mathbf{a} \in \mathcal{S}$, a guard zone $G(\mathbf{a}, \lambda)$ having an 'extent' λ is the subset of $\mathcal{S}$ defined by*

$$G(\mathbf{a}, \lambda) = \{\mathbf{x} : \delta(\mathbf{a}, \mathbf{x}) \leq \lambda\} \tag{2.1}$$

where

$$\lambda \geq 0.$$

Clearly, $G(\mathbf{a}, \lambda)$ is a hyperellipsoid in $\mathcal{S}$ with respect to the metric δ, whose size is controlled by a parameter λ, and which is centred at $\mathbf{a}$ (in $\mathcal{S}$).

In the subsequent discussions, unless stated otherwise, it will be assumed that $\mathcal{S} = \mathcal{R}^N$ and δ is the Mahalanobis distance [60], that is, it is defined by

$$\delta^2(\mathbf{x}, \mathbf{y}) = (\mathbf{x} - \mathbf{y})^T \mathbf{A} (\mathbf{x} - \mathbf{y}), \ \mathbf{x}, \mathbf{y} \in \mathcal{R}^N \tag{2.2}$$

$\mathbf{A}$ being a symmetric, positive definite $N \times N$ matrix.

Let $\mathbf{X}_1^{(k)}, \mathbf{X}_2^{(k)}, \mathbf{X}_3^{(k)}, \ldots$ be the sequence of learning (or training) samples for the kth class C_k. The following condition is assumed to hold:

(A5) The training samples for each class are independently distributed.

The generalized guard zones algorithm (GGA) for estimating $\theta^{(k)}$ recursively by $\hat{\theta}_n^{(k)}$ is as follows :

$$\hat{\theta}_n^{(k)} = \begin{cases} \mathbf{f}(\mathbf{X}_1^{(k)}) & \text{for } n = 1 \\ \hat{\theta}_{n-1}^{(k)} - a_n \mathbf{Y}_n^{(k)} & \text{for } n \geq 2 \end{cases} \tag{2.3}$$

where

$$\mathbf{Y}_n^{(k)} = \begin{cases} \hat{\theta}_{n-1}^{(k)} - \mathbf{f}(\mathbf{X}_n^{(k)}) & \text{if } \mathbf{X}_n^{(k)} \in G(\hat{\mu}_{n-1}^{(k)}, \lambda_n) \\ \mathbf{0} & \text{otherwise} \end{cases} \tag{2.4}$$

$\hat{\theta}_n^{(k)}$ is the nth-stage estimate of $\theta^{(k)}$

$\{a_n\}$: a sequence of positive numbers, with $a_n \leq 1$

$\mathbf{f}: \mathcal{R}^N \rightarrow \mathcal{R}^q$ is a continuous mapping, defining an unbiased statistic for $\theta^{(k)}$

$\hat{\mu}_{n-1}^{(k)}$: the $(n-1)$-th stage GGA estimate of $\mu^{(k)}$

$G(\hat{\mu}_{n-1}^{(k)}, \lambda_n)$ is the region $\{ \mathbf{x} : \mathbf{x} \in \mathcal{R}^N, d_n(\mathbf{x}, \hat{\mu}_{n-1}^{(k)}) \leq \lambda_n \}$

$d_k^2(\mathbf{x}, \mathbf{y})$ is the function $(\mathbf{x}-\mathbf{y})^T \mathbf{A}_n (\mathbf{x}-\mathbf{y})$

$\mathbf{A}_n$: a symmetric, positive definite matrix, which may or may not be a function of $\mathbf{X}_i^{(k)}$ and/or $\hat{\theta}_{n-1}^{(k)}$, $i = 1(1)n$

λ_n : a nonnegative number, prespecified

In essence, this algorithm uses only those training samples for updating the estimate, which lie within the corresponding guard zone centred at the preceding estimate of the mean. Training samples which lie outside it are ignored and the estimate kept unchanged at the corresponding stages.

If λ_n decreases progressively with n, the system gradually approaches the non-adaptive state, that is, no updating takes place eventually. Clearly, this is because the size of the guard zone and hence the probability of a training sample being from within the guard zone, decreases progressively with λ_n, so that the number of samples getting selected for the updating process decreases.

On the other hand, if the value of λ_n increases gradually with n, the system approaches the nonsupervised state, since the resulting progressive increase in the size of the guard zone makes the updating procedure less and less restrictive.

2.2.1 The non-GGA: its definition

The term 'non-GGA' will be used to refer to the usual stochastic approximation algorithm for estimating $\theta^{(k)}$ recursively under the setup defined earlier in this section. It can be defined as follows:

$$\hat{\hat{\theta}}_n^{(k)} = \begin{cases} \mathbf{f}(\mathbf{X}_1{}^{(k)}) & \text{for } n = 1 \\ \hat{\hat{\theta}}_{n-1}^{(k)} - a_n \tilde{\mathbf{Y}}_n^{(k)} & \text{for } n \geq 2 \end{cases} \tag{2.5}$$

where

$$\tilde{\mathbf{Y}}_n^{(k)} = \hat{\hat{\theta}}_{n-1}^{(k)} - \mathbf{f}(\mathbf{X}_n{}^{(k)}) \ , \tag{2.6}$$

all other symbols having the same significance as earlier in this chapter.

2.3 Relation of the GGA to the algorithms of Pal et al. and Chien

2.3.1 The algorithm of Pal et al. [53]

This algorithm is derived intuitively on the basis of the central assumption that the distribution of the members of a class in the feature space has a central tendency and that the probability of misclassification near these points of central tendency is substantially low. Thus it is possible to construct a region around the point of central tendency of a class, for which the probability of misclassification is so low that an unrestricted updating procedure for *the samples coming from such a region is highly likely to assist significantly in the convergence of the system.* They called such a region a *guard zone* (a term borrowed with gratitude by this work). They used a *fuzzy* classifier to obtain labels for training samples. Further, in defining a distance measure they assumed a diagonal form for the matrix $\mathbf{A}$ in equation 2.2, as follows:

$$\mathbf{A}_n = \begin{bmatrix} {\sigma_{1n}}^{-2} & 0 & 0 & \dots & 0 \\ 0 & {\sigma_{2n}}^{-2} & 0 & \dots & 0 \\ 0 & 0 & {\sigma_{3n}}^{-2} & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & {\sigma_{Nn}}^{-2} \end{bmatrix}$$

where σ_{jn} is the standard deviation of the jth feature in the nth class. Also, they assumed a constant value of λ_t for all classes and for all iterations t, that is, they assumed

$$\lambda_t = \lambda \quad \forall t.$$

For the particular problem of Telugu vowel classification with the first three formants as features, they were able to obtain empirical estimates for λ which were optimal in the sense of minimising the *mean square error*(MSE) of the performance scores of their algorithm with respect to those of the fully-supervised algorithm.

A point to note here is that their investigations were purely of an empirical nature. They did not attempt any sort of theoretical study of the behaviour of such algorithms, or some rigorous method for estimating the threshold value λ.

2.3.2 The algorithm of Chien [54]

This algorithm is derived in relation to a learning process in pattern recognition where steps must be taken to minimise the effect of *spurious* samples carrying unreliable information. With the assumption of multivariate normal density for each pattern class, a non-linear stochastic approximation-like algorithm was obtained, which reduces to discarding the spurious samples with a threshold element, while behaving as an ordinary linear algorithm in the meantime. The following assumptions regarding the composition of learning samples were made:

Assumption (i). Each learning sample $\mathbf{X}_i$ is composed of two parts:

$$\mathbf{X}_i = \mathbf{M} + \mathbf{N}_i, i = 1, 2, \ldots, n$$

where $\mathbf{N}_i$ is the noise component of the sample and $\mathbf{M}$ is the mean vector under estimation.

Assumption (ii). Each $\mathbf{N}_i$ is attributed to two distinct types of Gaussian noise. Type I noise can be thought of as the ordinary measurement variation that is inherent in the pattern samples for each class. This type of noise is associated with the genuine learning samples carrying reliable information for $\mathbf{M}$. Type II noise is associated with spurious learning samples that carry unreliable information in regard to the unknown $\mathbf{M}$. Let the type I noise be distributed in the multivariate normal form with mean vector $\mathbf{0}$ and dispersion matrix $\mathbf{K}_1$ and let the type II noise have an identical distribution but with a dispersion matrix $\mathbf{K}_2$. Further,

$$\mathbf{K}_1 = \mathbf{K}_2/\beta, \text{ where } \beta \gg 1.$$

Assumption (iii). For each $\mathbf{N}_i$, type I noise can occur with a probability $(1 - \alpha)$ and type II noise can occur with a probability α, where $\alpha \ll 1$.

Using the theory of random functions to minimise the mean square error of the estimate, Chien obtained the following nonlinear algorithm:

$$\mathbf{M}_n = \mathbf{M}_{n-1} + \mathbf{F}_n(\mathbf{X}_n - \mathbf{M}_{n-1}) \tag{2.7}$$

where $\mathbf{F}_n(\cdot)$ is a nonlinear function, a good approximation to which is:

$$\mathbf{F}_n(\mathbf{D}) = \begin{cases} \mathbf{K}_{n-1}\mathbf{K}_1{}^{-1}\mathbf{D} & \text{if } d(\mathbf{D}) \leq \theta_n \\ \mathbf{0} & \text{if } d(\mathbf{D}) > \theta_n \end{cases} \tag{2.8}$$

where

$$\mathbf{K}_{n-1} = \mathcal{E}[\mathbf{M}_{n-1} - \mathbf{M}][\mathbf{M}_{n-1} - \mathbf{M}]^T$$

and

$$d(\mathbf{D}) = \mathbf{D}^{\mathbf{T}}\mathbf{K_1}^{-1}\mathbf{D}.$$

This implies that learning samples that are found to be unreliable are simply discarded, while a linear transformation is carried out on the samples that seem to be reliable.

An estimate for the threshold θ_n was obtained as

$$\theta_n = 2 \ln \frac{1-\alpha}{\alpha} + \ln \beta$$

as the value for which the function $\mathbf{T}_n(\cdot)$ has a point of inflection, where $\mathbf{T}_n$ is such that

$$\mathbf{F}_n(\mathbf{D}) = \mathbf{T}_n(d(\mathbf{D}))\mathbf{D}.$$

Obtaining estimates for α and β is another problem, however.The error covariance $\mathbf{K}_{n-1}$ is estimated as follows:

If $\mathbf{M}_0$ is an initial guess for $\mathbf{M}$ with error dispersion matrix $\mathbf{K}_1/\lambda$, then

$$\mathbf{K}_{n-1} = [(n-1) + \lambda]^{-1}\mathbf{K}_1,$$

so that

$$\mathbf{F}_n(\mathbf{D}) = \begin{cases} [(n-1) + \lambda]^{-1}\mathbf{D} & \text{if } d(\mathbf{D}) \leq \theta_n \\ \mathbf{0} & \text{if } d(\mathbf{D}) > \theta_n \end{cases} \tag{2.9}$$

Thus the algorithm becomes equivalent to the GGA with

$$a_n = [(n-1) + \lambda]^{-1}.$$

2.4 Some intuitive considerations

It is possible to provide some intuitive justification for the class of learning algorithms typified by the GGA. The following discussion shows that such algorithms can be expected to converge in certain situations, and convergence, of course, is a highly desirable characteristic of a learning algorithm. In the following chapter, a rigorous proof for the stochastic convergence of the GGA is provided.

Keeping in mind the assumptions (A1)–(A5), let us consider the simple problem of estimating recursively the mean vector μ of a random variable $\mathbf{X}$. Let us consider the *worst possible* situation from the practical point of view, namely, that of $\mathbf{X}$ having a *uniform* distribution over the sample space Ω. If at any stage of learning, say, the nth, if the training sample $\mathbf{X}_n$ is such that it is

closer in some sense to μ than the preceding estimate $\hat{\mu}_{n-1}$ is, to μ, then by virtue of lemma 2.1 below it follows that the current estimate $\hat{\mu}_n$ obtained by taking a weighted average of $\mathbf{X}_n$ and $\hat{\mu}_{n-1}$, will be *closer* to μ than $\hat{\mu}_{n-1}$ is. If this happens at every stage of learning then the convergence of the estimates to the true value is assured. However, from lemma 2.2 below, it follows that for the type of distance defined in section 2.2, the condition stipulated above (namely, that $\mathbf{X}_n$ be *closer* to μ than $\hat{\mu}_{n-1}$ is) is implied by the condition that the distance between $\mathbf{X}_n$ and $\hat{\mu}_{n-1}$ be bounded above by a non-negative quantity $\ell(\hat{\mu}_{n-1}, \mathbf{X}_n, \mu)$. This, in essence, is nothing but the principle behind the GGA.

Lemma 2.1 *Let d be a metric defined over a convex metric space $\mathcal{C}$. If d satisfies the condition*

$$d(\omega\mathbf{x} + (1-\omega)\mathbf{y}, \mathbf{a}) \leq \omega d(\mathbf{x}, \mathbf{a}) + (1 - \omega d(\mathbf{y}, \mathbf{a})$$

$$\forall \omega \in [0,1], \quad \mathbf{x}, \mathbf{y}, \mathbf{a} \in \mathcal{C}$$

then

$$d(\mathbf{y}, \mathbf{a}) \leq d(\mathbf{x}, \mathbf{a})$$

implies that

$$d(\mathbf{z}, \mathbf{a}) \leq d(\mathbf{x}, \mathbf{a})$$

where

$$\mathbf{z} = \omega\mathbf{x} + (1-\omega)\mathbf{y}, \ \mathbf{z} \in \mathcal{C}$$

Lemma 2.2 *Let*

$$d^2{}_k(\mathbf{x}, \mathbf{y}) = (\mathbf{x}-\mathbf{y})'\mathbf{A}(\mathbf{x}-\mathbf{y}), \quad \mathbf{x}, \mathbf{y} \in \mathcal{R}^N,$$

$\mathbf{A}$ being a symmetric, positive definite $N \times N$ matrix. Then

$$d(\mathbf{y}, \mathbf{a}) \leq d(\mathbf{x}, \mathbf{a})$$

if and only if

$$d^2(\mathbf{x}, \mathbf{y}) \leq \ell(\mathbf{x}, \mathbf{y}, \mathbf{a})$$

where ℓ is a real-valued, non-negative, continuous map defined as

$$\ell(\mathbf{x}, \mathbf{y}, \mathbf{a}) = 2(\mathbf{x}-\mathbf{y})'\mathbf{A}(\mathbf{x}, \mathbf{a}).$$

2.5 The GGA as a robust statistical procedure

Generally speaking, robust statistics incorporates the theory and practice of a body of statistical procedures designed to deal with situations where the 'assumptions commonly made in statistics (such as normality, linearity, independence) are at most approximations to reality', according to Hampel et al. [57]. To quote them further:

> They (the deviations from assumptions) show up as outliers , which are far away from the bulk of the data, and are dangerous for many classical statistical procedures. (Barnett and Lewis [58] define an outlier in a set of data as *an observation (or subset of observations) which appears to be inconsistent with the remainder of that set of data*). The outlier problem is well known and probably as old as statistics, and any method for dealing with it, such as *subjective rejection* or *any formal rejection rule* belongs to robust statistics in this broad sense.

As the GGA is meant primarily to deal with non-ideal (recursive) estimation situations (that is, where there is a possibility of the presence of wrongly classified training samples) and is basically a rejection procedure, it easily qualifies as a robust statistical procedure. At this point, however, a few clarifications are necessary. The first is that while many agree [57,56,59] that any method, formal or informal, of dealing with outliers, which is reasonable and not totally inappropriate, prevents the worst, it is also the general consensus that 'the best rejection procedures do not quite reach the performance of the best robust procedures' [56]. It is a fact, however, that the rejection of outliers is a precursor of robust statistics, and as such is an integral part of it. The second point to be noted is that in most learning situations in pattern recognition, one is expected to work with multivariate data and, while the notion of an outlier can easily be carried over to the multivariate situation, one requires some *sub-ordering* principle in order to do so [58]. Thus, depending on the sub-ordering principle applied, there can be different types of outliers. The third point to note is that the GGA is required to do recursive estimation, for which there does not seem to have been much progress in designing robust methods [59].

So, at least in the particular context of recursive estimation of parameters of classifiers in pattern recognition, the use of a relatively unsophisticated robust technique like the GGA decidedly seems to have practical utility, though it need not be the most robust procedure under the circumstances.

2.6 Some remarks regarding the GGA

The proper selection of the various parameters of the GGA, namely, $\{a_n\}$, λ_n and $\mathbf{A}_n$, is understandably of crucial importance, so far as the efficiency of the estimates obtained is concerned. The choice can be based on some suitable criteria of 'goodness' of estimation procedures. For instance, if we insist that the algorithm should converge almost surely, then as shall be observed in chapter 3, a sufficient condition required to hold is

$$\sum_{n=1}^{\infty} {a_n}^2 < \infty.$$

Clearly then, a set of possible choices of a_n is

$$a_n = n^{-\delta},$$

where $\delta \in (\frac{1}{2}, \infty)$. In practice, it will be better to choose a small value of δ, or the corrections $\mathbf{Y}_n^{(k)}$ will be too small.

The choice of $\mathbf{A}_n$ is chiefly governed by the fact that it is used to define a distance function $d(\cdot, \cdot)$ which is as follows.

$$\delta^2(\mathbf{x}, \mathbf{y}) = (\mathbf{x} - \mathbf{y})^T \mathbf{A}(\mathbf{x} - \mathbf{y}), \ \mathbf{x}, \mathbf{y} \in \mathcal{R}^N$$

Statistical considerations dictate that, generally, $\mathbf{A}_n$ can have any one of the the following forms:

$$\mathbf{A}_n = \mathbf{I}_N, \text{ the identity matrix of order } N \tag{2.10}$$

or

$$\mathbf{A}_n = \begin{bmatrix} 1/s_{11}^{(n)} & 0 & 0 & \dots & 0 \\ 0 & 1/s_{22}^{(n)} & 0 & \dots & 0 \\ 0 & 0 & 1/s_{33}^{(n)} & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1/s_{NN}^{(n)} \end{bmatrix} \tag{2.11}$$

or

$$\mathbf{A}_n = \mathbf{S}_n^{-1} \tag{2.12}$$

where

$$\mathbf{S}_n = \left(\left(s_{ij}^{(n)} \right) \right),$$

$s_{ij}^{(n)}$ being the nth-stage estimate of the (i, j)th element of the estimated dispersion matrix.

The exact choice is generally governed by the nature of the underlying probability density of the feature vector in any given class, provided, of course, such information is available *a priori*. If the features can be expected to be uncorrelated and to have unit variances (for instance, if the values of the feature variables are normalised in some way), then the first choice is good enough. If however, they are uncorrelated but are not normalised, then the second choice is most suitable. In the most general situation, the last choice can be made.

The proper choice of the guard-zone parameter λ_n is the objective of the second half of this thesis. In section 3.7 of chapter 3 an attempt is made to obtain estimates on the basis of the criterion of stochastic convergence, while in chapter 4 estimates that optimise a performance index are obtained with the help of large-sample statistical distribution theory.

Chapter 3

Asymptotic and dynamic behaviour of the GGA

3.1 Introduction

The learning of unknown parameters of classifiers is an indispensible part of pattern recognition problems. If a sufficiently large set of correctly labeled training samples is available, then 'reasonably good' estimates of the parameters can generally be obtained, provided they converge in some sense to the true value, which is the goal of learning. As also mentioned in section 1.2, the goal of learning can actually be achieved only if the algorithms converge. In fact, a learning process can be considered to be successful if the attainment of the goal of learning can be guaranteed, that is, if the algorithms of learning converge. Generally, the goal of learning represents the state that the learning system *should reach* in the process of learning, and the selection of such a state is actually achieved by a proper choice of a certain functional that has an extremum which corresponds to that state. Convergence is generally defined in terms of one or more of the various modes of stochastic convergence found in statistical literature (for example,[60]), namely, almost sure convergence or mean- square convergence. These are defined in section 3.3, which also contains some auxiliary results from the theory of multidimensional stochastic approximation which will be used to prove various results concerning the convergence properties of the GGA and the non-GGA.

It was mentioned earlier on that if a sufficiently large set of correctly labeled training samples is available, then 'reasonably good' estimates of the parameters can generally be obtained. In many real-life situations, however, it is either difficult or expensive to obtain labels, for example, in remote sensing [61,62] and medical diagnosis [63,64,65,66], so that mislabeling of training samples can become one of the spectres with which a pattern recognition scientist has to contend. It is, therefore, useful to know how this problem can affect the learning procedure. A reasonable amount of work has been done for the two-class classification problem. The effects of random training errors on Fisher's discriminant function have been studied by Lachenbruch [63,64], McLachlan [67], Michalek and Tripathi [68], O'Neill [69], Krishnan [70,71], Chhikkara and McKeon [61] and Katre and Krishnan [72]. They concluded that the effect is to underestimate distance, overestimate error rate, introduce bias into estimates of the discriminant function,

make the maximum likelihood estimates of the discriminant function converge to nontrue values, and change the asymptotic relative efficiency (ARE) relative to a completely correctly classified sample of the same size. A good survey of this aspect of learning can be found in [73]. At this point, it may not be irrelevant to mention that in order to tackle this kind of problem in learning, quite a bit of work has been done. For instance, Dempster, Laird and Rubin [66] have suggested the use of the EM algorithm for this purpose, and more recently, Greblicki [74], Krishnan and Nandy [75] and Titterington [76] have advocated certain stochastic supervisors, while Chittineni [77,62] has developed some schemes for correcting labels. Incidentally, a related problem is the effect of correlated training samples on learning. Some research has been done in this field too, and a reasonably good bibliography can be found in [78,73,79,80].

In this chapter,the asymptotic behaviour of the GGA is investigated for the particular case in which errors occur in the labeling of training samples in an m-class N- feature pattern recognition problem. This is done under two different sets of conditions, namely:

- the situation where there is no mislabeling of training samples,
- the particular case in which errors occur in the labeling of training samples.

The effect of mislabeling is to cause 'wrong' samples to be used in the recursive learning of the estimates, for any given class. A simple but realistic model [77] is adopted to describe this sort of situation,and is discussed in section 3.2. Under this model, the stochastic convergence of the class of recursive learning procedures that the GGA represents is investigated. It is found that, under certain conditions, these estimates do converge strongly, that is, with probability one, but to nontrue values, more specifically, to convex linear combinations of true parameters of all m classes. This conclusion is reached using some results on multidimensional stochastic approximation [81]. A detailed proof is provided in section 3.4 for the ideal case of no mislabeling, and in section 3.6 for the other, more general situation involving mislabeled training samples. As a matter of academic interest, similar results regarding the stochastic convergence of the non-GGA are proved in section 3.5. The results obtained, in themselves, are not surprising, because the presence of mislabeled samples in the training set is sure to affect the behavior of the training process in some way. This work merely provides a mathematical description of the effect on its convergence.

Section 3.7 deals with a slightly different aspect of the GGA. While examining the conditions obtained in the earlier sections for the convergence of the algorithm, it is found that the choice of the guard-zone parameter for which convergence is highly likely to be assured, is possibly a function of the values of the current training samples and the past estimate. In other words, the value of $\lambda_n^{(k)}$ differs from iteration to iteration, and adapts itself to the existing situation, so to speak. This characteristic of the GGA is described by the adjective **dynamic**. In this context, some bounds (lower and upper) for $\lambda_n^{(k)}$ are obtained and certain convex linear combinations of the two bounds are suggested as estimates for $\lambda_n^{(k)}$.

3.2 A model for labeling errors [77]

The model used to describe the situation where there is a possibility of the training samples being mislabeled,was developed by Chittineni [77]. It can be specified as follows. Let ω and $\hat{\omega}$ denote, respectively, the true and the given labels of the training samples $\mathbf{X}_i$, i=1,2,.... Clearly,

$$\omega\ ,\hat{\omega}\ \in\{1,2,\ldots,m\}.$$

Also, in terms of the notation $\mathbf{X}_i^{(k)}$ used earlier on to denote the ith training sample for the kth class,

$$\mathbf{X}_i^{(k)} \equiv [\mathbf{X}_i \mid \hat{\omega}\ = k]\ \ \forall i.$$

Let $\pi_i = P(\omega\ = i)$ denote the *a priori* probability for the ith class C_i. Further, let $p_i(\mathbf{X}) = p(\mathbf{X} \mid \omega\ = i)$ be the class-conditional density of the feature vector $\mathbf{X}$ for C_i. Also, let α_{ij} denote the probability that a sample from C_j has been given the label i, that is,

$$\alpha_{ij} = P(\hat{\omega}\ = i \mid \omega\ = j),\quad i,j = 1,2,\ldots,m \tag{3.1}$$

Clearly,

$$\sum_{i=1}^{m}\alpha_{ij} = 1, \tag{3.2}$$

that is,

$$\mathbf{A}^T_{m\times m}\epsilon_{m\times 1} = \epsilon_{m\times 1}$$

where

$$\epsilon_{m\times 1} = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}$$

and

$$\mathbf{A} = ((\alpha_{ij}))$$

Now,

$$\begin{aligned} p(\mathbf{X} \mid \hat{\omega}\ = i) &= \frac{p(\mathbf{X},\hat{\omega}\ = i)}{p(\hat{\omega}\ = i)} \\ &= \frac{1}{p(\hat{\omega}\ = i)}\sum_{j=1}^{m} p(\mathbf{X},\hat{\omega}\ = i,\omega\ = j) \\ &= \frac{1}{p(\hat{\omega}\ = i)}\sum_{j=1}^{m} p(\mathbf{X} \mid \hat{\omega}\ = i,\omega\ = j)P(\hat{\omega}\ = i\mid,\omega\ = j)P(\omega\ = j) \\ &= \frac{1}{p(\hat{\omega}\ = i)}\sum_{j=1}^{m} \pi_j\alpha_{ij}p(\mathbf{X} \mid \hat{\omega}\ = i,\omega\ = j) \end{aligned}$$

$$= \sum_{j=1}^{m} \epsilon_{ij} p(\mathbf{X} \mid \hat{\omega} = i, \omega = j) \tag{3.3}$$

where

$$\epsilon_{ij} = \frac{\pi_j \alpha_{ij}}{\sum_{k=1}^{m} \pi_k \alpha_{ik}} \tag{3.4}$$

since

$$\begin{aligned} P(\hat{\omega} = i) &= \sum_{k=1}^{m} P(\hat{\omega} = i, \omega = k) \\ &= \sum_{k=1}^{m} P(\hat{\omega} = i \mid \omega = k) P(\omega = k) \\ &= \sum_{k=1}^{m} \pi_k \alpha_{ik} \end{aligned}$$

If it is possible to assume that

(A6) $p(\mathbf{X} \mid \omega = j) = p(\mathbf{X} \mid \hat{\omega} = i, \omega = j) \quad \forall i, j$

then the equation 3.3 becomes

$$\begin{aligned} p(\mathbf{X} \mid \hat{\omega} = i) &= \sum_{j=1}^{m} \epsilon_{ij} p(\mathbf{X} \mid \omega = j) \\ &= \sum_{j=1}^{m} \epsilon_{ij} p_j(\mathbf{X}) \end{aligned}$$

It may not be out of place to emphasize here that assumption **(A6)** is perfectly reasonable in the sense that it merely requires that the distribution of $\mathbf{X}$ in any class depend not on the given label $\hat{\omega}$, but only on the true label ω .

Further, the following lemma can be shown to be true.

Lemma 3.1 *Given the setup defined in this section, for any subset $A_k(n)$ of the sample space, the probability density of $\mathbf{X}_n^{(k)}$, that is, a sample labelled k at the nth stage, can be rewritten as*

$$\begin{aligned} p(\mathbf{X}_n^{(k)}) &= p(\mathbf{X}_n \mid \hat{\omega} = k) \\ &= P(\mathbf{X} \mid \hat{\omega} = k) \\ &= \begin{cases} \sum_{j=1}^{m} \frac{\beta_{kj}(n)}{P(A_k(n) \mid \hat{\omega} = k)} p(\mathbf{X} \mid \omega = j) & \text{if } \mathbf{X}_n^{(k)} \in A_k(n), \text{ given } \hat{\omega} = k \\ \sum_{j=1}^{m} \frac{\beta_{kj}(n)}{P(A_k(n)^c \mid \hat{\omega} = k)} p(\mathbf{X} \mid \omega = j) & \text{otherwise} \end{cases} \end{aligned} \tag{3.5}$$

where

$$\beta_{kj}(n) = P(A_k(n)|\mathbf{X}, \hat{\omega} = k, \omega = j)\epsilon_{kj} \tag{3.6}$$

$$\beta_{kj}^*(n) = P(A_k(n)^c|\mathbf{X}, \hat{\omega} = k, \omega = j)\epsilon_{kj} \tag{3.7}$$

provided we are prepared to assume that

(A6) $p(\mathbf{X}|\hat{\omega} = k, \omega = j) = p(\mathbf{X}|\omega = j) \quad \forall j, k = 1, 2, \ldots, m$
(A7) $P(\hat{\omega} = k, A_k(n)) > 0 \ \forall k = 1, 2, \ldots, m \text{ and } n \geq 1$
(A8) $P(\hat{\omega} = k, A_k(n)^c) > 0 \ \forall k = 1, 2, \ldots, m \text{ and } n \geq 1$

Proof: From well-known results in probability theory, it follows that

$$p(\mathbf{X}|\hat{\omega} = k) = \begin{cases} p(\mathbf{X}|\hat{\omega} = k, A_k(n)) & \text{if } \mathbf{X} \in A_k(n) \text{ given } \hat{\omega} = k \\ p(\mathbf{X}|\hat{\omega} = k, A_k(n)^c) & \text{otherwise} \end{cases}$$

where $A_k(n)^c$ denotes the event *complementary* to $A_k(n)$ in the feature space $\Omega_{\mathbf{X}}$. However,

$$p(\mathbf{X}|\hat{\omega} = k, A_k(n)) \tag{3.8}$$

$$= \frac{p(\mathbf{X}, \hat{\omega} = k, A_k(n))}{P(\hat{\omega} = k, A_k(n))}$$

$$= \frac{\sum_{j=1}^m p(\mathbf{X}, \hat{\omega} = k, A_k(n), \omega = j)}{P(\hat{\omega} = k, A_k(n))}$$

$$= \frac{1}{P(\hat{\omega} = k, A_k(n))} \times$$

$$\sum_{j=1}^m P(A_k(n)|\mathbf{X}, \hat{\omega} = k, \omega = j)p(\mathbf{X}|\hat{\omega} = k, \omega = j) \times$$

$$P(\hat{\omega} = k, \omega = j)P(\omega = j)$$

$$= \frac{1}{P(A_k(n)|\hat{\omega} = k)P(\hat{\omega} = k)} \times$$

$$\sum_{j=1}^m P(A_k(n)|\mathbf{X}, \hat{\omega} = k, \omega = j)\alpha_{kj}\pi_j p(\mathbf{X}|\hat{\omega} = k, \omega = j)$$

$$= \frac{1}{P(A_k(n)|\hat{\omega} = k)} \sum_{j=1}^m \beta_{kj}(n)p(\mathbf{X}|\omega = j) \tag{3.9}$$

by the assumptions (A6) and (A7) and since

$$P(\hat{\omega} = k) = \sum_{j=1}^m \alpha_{kj}\pi_j.$$

Similarly, we must have

$$p(\mathbf{X}|\hat{\omega} = k, A_k(n)^c) = \frac{1}{P(A_k(n)^c|\hat{\omega} = k)} \beta_{kj}{}^*(n) p(\mathbf{X}|\omega = j) \qquad (3.10)$$

in view of the assumptions (A6) and (A8).

This proves the lemma.

It is not difficult to observe that the quantities $\beta_{kj}(n)$ and $\beta_{kj}{}^*(n) \in [0,1]\ \forall k, j = 1, 2, \ldots, m$.

We have not studied the problem of estimating the mislabeling probabilities α_{kj} yet. Off-hand, it can be said, however, that they can be estimated if some measures of the probability of error of the labelling process involved are available. For instance, if the labelling is done with the help of some statistical classifier, then the error can be measured by its probabilities of misclassification, provided these can be estimated with sufficient accuracy.

3.3 Stochastic convergence of learning algorithms

The stochastic convergence of a recursive discrete algorithm for estimating a parameter θ by $\hat{\theta}_n$, can be defined in various ways. For instance, we say that

(a) the sequence $\{\hat{\theta}_n\}$ converges to θ **with probability one** or **almost surely** if

$$P\left(\lim_{n\to\infty} \| \hat{\theta}_n - \theta \| = 0\right) = 1$$

(b) $\hat{\theta}_n$ converges to θ **in the mean-square** sense if

$$\lim_{n\to\infty} \mathcal{E}[\| \hat{\theta}_n - \theta \|^2] = 0,$$

$\mathcal{E}$ being the expectation operator.

Throughout this chapter,extensive use will be made of the following results, due to Schmetterer [81].

Lemma 3.2 *Let $\{a_n\}$ be a sequence of positive real numbers such that*

*(**B1**)* $\sum_{n=1}^{\infty} a_n{}^2 < \infty$.

Let $\mathbf{x_n}$ and $\mathbf{y_n}$ be k-dimensional random vectors that satisfy

*(**B2**)* $\mathbf{x_{n+1}} = \mathbf{x_n} - a_n \mathbf{y_n}, n \geq 1$.

Let $\mathbf{M}_n$ be a measureable mapping from R^k to $\mathcal{R}^k$ such that

*(**B3**)* $\mathcal{E}(\mathbf{y_n} \mid \mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_n}) = \mathbf{M}_n(\mathbf{x_n}) a.e.$

Let a,b,c be nonnegative real numbers, and let

*(**B4**)* $\mathcal{E}(\| \mathbf{y_n} \|^2 \mid \mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_n}) \leq a + b \| \mathbf{x}_n \| + c \| \mathbf{x}_n \|^2 a.e.$

Also, for every $\mathbf{x} \in \mathcal{R}^k$ *and* $n \geq 1$,

(B5) $\mathbf{x}^T \mathbf{M}_n \mathbf{x} \geq 0$.

If $\mathbf{x}_1$ *is chosen in such a way that*

(B6) $\mathcal{E}(\| \mathbf{x}_1 \|^2)$ *exists,*

then the sequence $\{\mathbf{x}_n\}$ *converges with probability 1, that is, almost surely and the sequence* $\mathcal{E}(\| \mathbf{x}_n \|^2)$ *converges also.*

Lemma 3.3 *Suppose that conditions (B1)-(B6) hold. If, further, there exists for every* $\eta > 0$ *a* $\delta > 0$ *such that for* $n \geq 1$,

(B7) $\inf_{\eta < \|\mathbf{x}\| < \eta^{-1}} \mathbf{x}^T \mathbf{M}_n(\mathbf{x}) \geq \delta$,

then $\mathbf{x}_n$ *converges almost surely to the k-dimensional null vector* $\mathbf{0}$.

3.4 Convergence of the GGA in the ideal case [4]

Proposition 3.1 *For the problem of estimating* $\theta^{(k)}$ *recursively under the setup defined in section 2.2, by the assumptions (A1)- (A6), let* $\hat{\theta}_n^{(k)}$ *be the sequence of estimates, where* $\hat{\theta}_n^{(k)}$ *is given by the equation 2.3. If*

(C1) $\sum_{n=1}^{\infty} {a_n}^2 < \infty$

(C2) $p_n{}^{(k)} = P\left(d_n(\hat{\mu}_{n-1}^{(k)}, \mathbf{X}_n{}^{(k)}) \leq \lambda_n \mid \hat{\mu}_{n-1}^{(k)}\right) > \delta \quad \forall n$ *for some* $\delta > 0$

(C3) all the training samples are correctly labeled

(C4) the statistic $\mathbf{f}(\mathbf{X})$ *admits of second-order moments, with*

$$\mathcal{E}\left[\| \mathbf{f}(\mathbf{X}) \|^2 \mid \omega = i\right] = \rho_i$$

then

(a) $\{\hat{\theta}_n^{(k)}\}$ *converges with probability 1 to* $\theta^{(k)}$ *as* $n \to \infty$;

(b) $\{\mathcal{E}\| \hat{\theta}_n^{(k)} - \theta^{(k)} \|^2\}$ *converges as* $n \to \infty$.

Proof. Writing $\phi_n{}^{(k)} = \hat{\theta}_n^{(k)} - \theta^{(k)}$, we have

$$\phi_n{}^{(k)} = \begin{cases} \mathbf{f}(\mathbf{X}_1{}^{(k)}) - \theta^{(k)} & \text{for } n = 1 \\ \phi_{n-1}{}^{(k)} - a_n \mathbf{Z}_n{}^{(k)} & \text{for } n \geq 2 \end{cases} \tag{3.11}$$

where

$$\mathbf{Z}_n{}^{(k)} = \begin{cases} \phi_{n-1}{}^{(k)} - (\mathbf{f}(\mathbf{X}_n{}^{(k)}) - \theta^{(k)}) & \text{if } \mathbf{X}_n{}^{(k)} \in G(\hat{\mu}_{n-1}^{(k)}, \lambda_n) \\ \mathbf{0} & \text{otherwise} \end{cases} \tag{3.12}$$

This is because it is possible to write

$$\theta^{(k)} = (1 - a_n)\theta^{(k)} + a_n\theta^{(k)}$$

and $0 < a_n \leq 1$ by choice.

Thus the propositions can be shown to be true if it can be shown that under the conditions assumed therein,

(i) $\phi_n^{(k)}$ converges almost surely to **0** as $n \to \infty$

(ii) $\{\mathcal{E}[\| \phi_n^{(k)} \|^2]\}$ converges as $n \to \infty$.

To establish these, the lemmas 3.2 and 3.3 are applied directly, by showing that conditions (B1)-(B7) are true for $\phi_n^{(k)}$ as defined by equations 3.11 and 3.12.

The condition (B1) is true because of the assumption (C1), and the condition (B2) is seen to be hold in view of equations 3.11 and 3.12.

Now,

$$\begin{aligned}
&\mathcal{E}[\mathbf{Z}_n^{(k)} \mid \phi_1^{(k)}, \phi_2^{(k)}, \ldots, \phi_n^{(k)}] \\
&= p_{n+1}^{(k)}\mathcal{E}[\phi_n^{(k)} - \mathbf{f}(\mathbf{X}_{n+1}^{(k)}) - \theta^{(k)} \mid \phi_1^{(k)}, \phi_2^{(k)}, \ldots, \phi_n^{(k)}] \\
&= p_{n+1}^{(k)}[\phi_n^{(k)} - \mathcal{E}(\mathbf{f}(\mathbf{X}_{n+1}^{(k)}) - \theta^{(k)})] \\
&\quad \text{as } \mathbf{X}_{n+1}^{(k)} \text{ is independent of } \mathbf{X}_1^{(k)}, \mathbf{X}_2^{(k)}, \ldots, \mathbf{X}_n^{(k)} \\
&\quad \text{and hence } \phi_1^{(k)}, \phi_2^{(k)}, \ldots, \phi_n^{(k)}, \text{ by the assumption (A5)} \\
&= p_{n+1}^{(k)}\phi_n^{(k)} \text{ as } \mathbf{f}(\mathbf{X}^{(k)}) \text{ is unbiased for } \theta^{(k)} \\
&\quad \text{and so } \mathcal{E}(\mathbf{f}(\mathbf{X}_{n+1}^{(k)})) = \theta^{(k)} \\
&\quad \text{since the } \mathbf{X}_i^{(k)}\text{'s, i=1,2,... are identically distributed,by (C3)}
\end{aligned}$$

This verifies the condition (B3), with

$$\mathbf{M}_n(\mathbf{x}) = p_{n+1}^{(k)}\mathbf{x}, \ \mathbf{x} \in \mathcal{R}^N.$$

Further,

$$\begin{aligned}
&\mathcal{E}\left[\| \mathbf{Z}_n^{(k)} \|^2 \mid \phi_1^{(k)}, \phi_2^{(k)}, \ldots, \phi_n^{(k)}\right] \\
&= p_{n+1}^{(k)}\mathcal{E}\left[\| \phi_n^{(k)} - (\mathbf{f}(\mathbf{X}_{n+1}^{(k)}) - \theta^{(k)}) \|^2 \mid \phi_1^{(k)}, \phi_2^{(k)}, \ldots, \phi_n^{(k)}\right] \\
&= p_{n+1}^{(k)}\left[\| \phi_n^{(k)} \|^2 - 2\phi_n^{(k)\,T}\mathcal{E}(\mathbf{f}(\mathbf{X}_{n+1}^{(k)}) - \theta^{(k)})\right.
\end{aligned}$$

$$+ \mathcal{E}\| \mathbf{f}(\mathbf{X}_{n+1}{}^{(k)}) - \theta^{(k)^2}\Big]$$

as $\mathbf{X}_{n+1}{}^{(k)}$ is independent of $\mathbf{X}_1{}^{(k)}, \mathbf{X}_2{}^{(k)}, \ldots, \mathbf{X}_n{}^{(k)}$

and hence $\phi_1{}^{(k)}, \phi_2{}^{(k)}, \ldots, \phi_n{}^{(k)}$, by the assumption (A5)

$$= p_{n+1}{}^{(k)} \Big[\| \phi_n{}^{(k)} \|^2 - \| \theta^{(k)} \|^2 + \mathcal{E}[\| \mathbf{f}(\mathbf{X}^{(k)}) \|^2]\Big]$$

as the $\mathbf{X}_i{}^{(k)}$'s are independently and identically distributed

(by (A5) and (C3) respectively) and $\mathcal{E}\mathbf{f}(\mathbf{X}^{(k)}) = \theta^{(k)}$

$$\leq \| \phi_n{}^{(k)} \|^2 + \rho_k \text{ by (C4), as } p_{n+1}{}^{(k)} \leq 1$$

and $\| \theta^{(k)} \|^2 \leq \mathcal{E}[\| \mathbf{f}(\mathbf{X}^{(k)}) \|^2]$, as $\mathcal{E}\mathbf{f}(\mathbf{X}^{(k)}) = \theta^{(k)}$

which means that the condition (B4) holds with

$$a = \rho_k, \quad b = 0, \quad c = 1.$$

Also,

$$\mathbf{x}^T \mathbf{M}^n(\mathbf{x}) = p_{n+1}{}^{(k)} \mathbf{x}^T \mathbf{x} \geq 0 \ \forall \mathbf{x} \in \mathcal{R}^N,$$

which verifies the condition (B5).

That (B6) holds, is rather obvious, in view of (C4), as

$$\mathcal{E}\| (\mathbf{X}_1{}^{(k)}) - \theta^{(k)} \|^2 = \mathcal{E}[\| \mathbf{f}(\mathbf{X}^{(k)}) \|^2] - \| \theta^{(k)} \|^2$$

Finally, we can see that by virtue of the assumption (C2), the condition (B7) holds, for

$$\inf_{\eta < \|\mathbf{x}\| < \eta^{-1}} \mathbf{x}^T \mathbf{M}_n(\mathbf{x})$$

$$= \inf_{\eta < \|\mathbf{x}\| < \eta^{-1}} p_{n+1}{}^{(k)} \mathbf{x}^T \mathbf{x}$$

$$> \delta \eta^2$$

$$> 0$$

Thus the lemmas 3.2 and 3.3 hold for $\hat{\theta}_n{}^{(k)}$. Hence the proposition is proved.

3.5 Convergence of the non-GGA in the non-ideal case [5,6]

It is possible to prove the following result regarding the convergence of the non-GGA under the model adopted for describing errors in the labeling of training samples, given in section 3.2.

Proposition 3.2 *For the problem of estimating* $\theta^{(k)}$ *recursively under the setup defined in section 2.2, by the assumptions (A1)- (A6), let* $\hat{\hat{\theta}}_n^{(k)}$ *be the sequence of estimates used, where* $\hat{\hat{\theta}}_n^{(k)}$ *is given by the equation 2.5. If there is a possibility of training samples being mislabeled, and the model assumed in section 3.2 is taken to be valid, and if*

(CNG1) $\sum_{n=1}^{\infty} a_n^2 < \infty$

and

(CNG2) the statistic $\mathbf{f(X)}$ *admits of second-order moments, with*

$$\mathcal{E}\left[\| \mathbf{f(X)} \|^2 \mid \omega = i\right] = \rho_i$$

then

(a) $\{ \hat{\hat{\theta}}_n^{(k)} \}$ *converges with probability 1 to* $\bar{\theta}^{(k)}$ *as* $n \to \infty$;

(b) $\{\mathcal{E}\| \hat{\hat{\theta}}_n^{(k)} - \bar{\theta}^{(k)} \|^2 \}$ *converges as* $n \to \infty$,

where

$$\bar{\theta}^{(k)} = \sum_{j=1}^{m} \epsilon_{kj} \theta^{(j)}, \tag{3.13}$$

where ϵ_{kj} *is as in equation 3.4, that is,*

$$\epsilon_{kj} = \frac{\pi_j \alpha_{kj}}{\sum_{i=1}^{m} \pi_i \alpha_{ki}}$$

Proof: The validity of the proposition can be inferred directly from lemmas 3.2 and 3.3, provided one can show that the conditions (B1)-(B7) mentioned therein hold for $\psi_n^{(k)}$, where

$$\psi_n^{(k)} = \hat{\hat{\theta}}_n^{(k)} - \bar{\theta}^{(k)}.$$

It follows from equation 2.5 that for $k = 1, 2, \ldots, m$

$$\psi_{n+1}^{(k)} = \begin{cases} \mathbf{g}(\mathbf{X}_1^{(k)}) & \text{for } n = 0 \\ \psi_n^{(k)} - a_{n+1} \mathbf{Z}_{n+1}^{*} & \text{otherwise} \end{cases} \tag{3.14}$$

where

$$\mathbf{Z}_{n+1}^{*} = \psi_n^{(k)} - \left[\mathbf{f}(\mathbf{X}_{n+1}^{(k)}) - \bar{\theta}^{(k)}\right] \tag{3.15}$$

The condition (B1) is seen to be true because of (CNG1), and (B2) is equivalent to equation 3.14. The requirement (B3) is also satisfied, with

$$\mathbf{M}_n(\mathbf{x}) = \mathbf{x}\,,$$

as

$$\begin{aligned}
&\mathcal{E}\left[\mathbf{Z}_{n+1}{}^{*} \mid \psi_{\ 1}{}^{(k)}, \psi_{\ 2}{}^{(k)}, \ldots, \psi_{\ n}{}^{(k)}\right] \\
&= \psi_{\ n}{}^{(k)} - \mathcal{E}[\mathbf{f}(\mathbf{X}_n{}^{(k)})] + \bar{\bar{\theta}}^{\,(k)} \\
&\quad \text{since } \mathbf{X}_{n+1}{}^{(k)} \text{ is independent of } \mathbf{X}_1{}^{(k)}, \mathbf{X}_2{}^{(k)}, \ldots, \mathbf{X}_n{}^{(k)} \\
&\quad \text{and hence of } \psi_{\ 1}{}^{(k)}, \psi_{\ 2}{}^{(k)}, \ldots, \psi_{\ n}{}^{(k)} \\
&= \psi_{\ n}{}^{(k)} - \mathcal{E}[\mathbf{f}(\mathbf{X}) \mid \hat{\omega} = k] + \bar{\bar{\theta}}^{\,(k)} \\
&= \psi_{\ n}{}^{(k)}
\end{aligned}$$

as, by the equation 3.3,

$$\mathcal{E}[\mathbf{f}(\mathbf{X}) \mid \hat{\omega} = k] = \sum_{j=1}^{n} \epsilon_{kj} \theta^{\,(j)} = \bar{\bar{\theta}}^{\,(k)}\,. \tag{3.16}$$

Similarly, we have

$$\begin{aligned}
&\mathcal{E}\left[\| \mathbf{Z}_{n+1}{}^{*} \|^2 \mid \psi_{\ 1}{}^{(k)}, \psi_{\ 2}{}^{(k)}, \ldots, \psi_{\ n}{}^{(k)}\right] \\
&= \mathcal{E}\left[\| \mathbf{Z}_{n+1}{}^{*} \|^2\right] \text{ by the condition (A5)} \\
&= \| \psi_{\ n}{}^{(k)} \|^2 + \mathcal{E}\left[\| \mathbf{f}(\mathbf{X}) - \bar{\bar{\theta}}^{\,(k)} \|^2 \mid \hat{\omega} = k\right] \text{ by the equation 3.16} \\
&\leq \| \psi_{\ n}{}^{(k)} \|^2 + \mathcal{E}\left[\| \mathbf{f}(\mathbf{X}) \|^2 \mid \hat{\omega} = k\right] \\
&\quad \text{as } \mathcal{E}\left[\| \mathbf{f}(\mathbf{X}) \|^2 \mid \hat{\omega} = k\right] \geq \mathcal{E}\left[\| \mathbf{f}(\mathbf{X}) - \bar{\bar{\theta}}^{\,(k)} \|^2 \mid \hat{\omega} = k\right] \\
&= \| \psi_{\ n}{}^{(k)} \|^2 + \sum_{j=1}^{m} \epsilon_{kj} \rho_j \text{ by the condition (CNG2) and the equation 3.3} \\
&\leq \| \psi_{\ n}{}^{(k)} \|^2 + \sum_{j=1}^{m} \rho_j \quad \text{as } \epsilon_{kj} \leq 1 \forall k, j
\end{aligned}$$

Thus the requirement (B4) is seen to be satisfied with

$$a = \sum_{j=1}^{m} \rho_j, \quad b = 0, \quad c = 1.$$

The requirement (B5) is seen to be met as

$$\mathbf{x}\mathbf{M}_n(\mathbf{x}) = \mathbf{x}^T\mathbf{x} \geq 0.$$

The condition (B6)is satisfied , because

$$\begin{aligned}
&\mathcal{E}\left[\| \psi_1^{(k)} \|^2\right] \\
&= \mathcal{E}\left[\| \mathbf{f}(\mathbf{X}) - \bar{\bar{\theta}}^{(k)} \|^2 \mid \hat{\omega} = k\right] \\
&\leq \mathcal{E}\left[\| \mathbf{f}(\mathbf{X}) \|^2 \mid \hat{\omega} = k\right] \\
&= \sum_{j=1}^{m} \epsilon_{kj}\rho_j \\
&< \infty
\end{aligned}$$

Finally, (B7) follows because

$$\begin{aligned}
&\inf_{\eta<\|\mathbf{x}\|<\eta^{-1}} \mathbf{x}^T\mathbf{M}_n(\mathbf{x}) \\
&= \inf_{\eta<\|\mathbf{x}\|<\eta^{-1}} \mathbf{x}^T\mathbf{x} \\
&> \eta^2 \\
&> 0
\end{aligned}$$

Hence the proposition.

Implications of the proposition

(a) If the matrix $\mathbf{A}$ is equal to $\mathbf{I}_m$, the identity matrix of order m, i.e., if there is no mislabeling then under our assumptions,

$$\hat{\theta}_n^{(k)} \xrightarrow{a.s.} \theta^{(k)}$$

as expected.

(b) If $\mathbf{A} \neq \mathbf{I}_m$, then clearly the estimates for the different classes converge to nontrue values

$$\textstyle\sum_{j=1}^{m} \epsilon_{kj}\theta^{(j)}$$

i.e., a convex linear combination of the parameter vectors of all the classes, as it follows directly from the equation 3.4 that

$$\sum_{j=1}^{m} \epsilon_{ij} = 1 \ \forall i = 1, 2, \ldots, m. \tag{3.17}$$

(c) Yet another implication can be stated formally as follows.

Proposition 3.3 *For the problem of estimating* $\theta^{(k)}$ *recursively under the setup defined in section 2.2, by the assumptions (A1)- (A6), let* $\hat{\hat{\theta}}_n^{(k)}$ *be the sequence of estimates used, where* $\hat{\hat{\theta}}_n^{(k)}$ *is given by the equation 2.5. If there is a possibility of training samples being mislabeled, and the model assumed in section 3.2 is taken to be valid, and if*

(CNG1) $\sum_{n=1}^{\infty} a_n{}^2 < \infty$
and
(CNG2) the statistic $\mathbf{f(X)}$ *admits of second-order moments, with*

$$\mathcal{E}\left[\| \mathbf{f(X)} \|^2 \mid \omega = i\right] = \rho_i$$

then

$$\sum_{j=1}^{m} \gamma_{kj} \hat{\theta}_n^{(k)} \xrightarrow{a.s.} \theta^{(k)}$$

where

$$\Gamma = ((\gamma_{ij}))$$

is a generalized inverse [82] of the matrix

$$\mathbf{E} = ((\epsilon_{ij}))$$

satisfying

$$\mathbf{E}\,\Gamma = \mathbf{I}_m. \tag{3.18}$$

Proof: Firstly, we note that the matrix E is not full- rank as shown by the equation 3.17. Consequently,

$$\text{rank}(\mathbf{E}) = r \leq m - 1.$$

From proposition 3.2, it is known that if $\mathbf{E}^T$ denotes the transpose of E , then

$$\left(\hat{\theta}_n^{(1)} \middle| \hat{\theta}_n^{(2)} \middle| \cdots \middle| \hat{\theta}_n^{(m)}\right) \xrightarrow{a.s.} \mathbf{E}^T \left(\theta^{(1)} \middle| \theta^{(2)} \middle| \cdots \middle| \theta^{(m)}\right) \text{ element-wise}$$

(i.e., every *element* of the matrix on the left-hand side converges almost surely to the corresponding *element* on the right-hand side). By well-known results on almost sure convergence it follows that

$$\left(\hat{\theta}_n^{(1)} \middle| \hat{\theta}_n^{(2)} \middle| \cdots \middle| \hat{\theta}_n^{(m)}\right) \xrightarrow{a.s.} \mathbf{E}^T \left(\theta^{(1)} \middle| \theta^{(2)} \middle| \cdots \middle| \theta^{(m)}\right) \text{ column-wise}$$

(i.e., every *column* of the matrix on the left-hand side converges a.s. to the corresponding *column* on the right-hand side), and hence

$$\Gamma^T \left(\hat{\theta}_n^{(1)} \middle| \hat{\theta}_n^{(2)} \middle| \cdots \middle| \hat{\theta}_n^{(m)}\right) \xrightarrow{a.s.} \left(\theta^{(1)} \middle| \theta^{(2)} \middle| \cdots \middle| \theta^{(m)}\right) \text{ column-wise}$$

3.6 Convergence of the GGA in the non-ideal case [7]

Proposition 3.4 *For the problem of estimating* $\theta^{(k)}$ *recursively under the setup defined in section 2.2, by the assumptions (A1)- (A6), let* $\hat{\theta}_n^{(k)}$ *be the sequence of estimates, where* $\hat{\theta}_n^{(k)}$ *is given by the equation 2.3. If there is a possibility of training samples being mislabeled, the model assumed in section 3.2 being taken to be valid, and if*

(CG1) $\sum_{n=1}^{\infty} a_n^2 < \infty$

(CG2) $p_n^{(k)} = P(A_k(n)|\hat{\omega} = k)$

$$= P\left(d_n(\hat{\mu}_{n-1}^{(k)}, \mathbf{X}_n^{(k)}) \leq \lambda_n \mid \hat{\mu}_{n-1}^{(k)}, \hat{\omega} = k\right)$$
$$> \delta \quad \forall n \text{ for some } \delta > 0$$

(CG3) *the statistic* $\mathbf{f(X)}$ *admits of second-order moments, with*

$$\mathcal{E}\left[\| \mathbf{f(X)} \|^2 \mid \omega = i\right] = \rho_i$$

then

1. $\{ \hat{\theta}_n^{(k)} - \theta_n^{(k)} \}$ *converges with probability 1 to* $\mathbf{0}$ *as* $n \to \infty$, *where*

$$\bar{\theta}_n^{(k)} = \sum_{j=1}^{m} \beta_{kj}(n)\theta^{(j)}, \tag{3.19}$$

2. $\{\mathcal{E}\| \hat{\theta}_n^{(k)} - \bar{\theta}_n^{(k)} \|^2\}$ *converges as* $n \to \infty$.

Proof. Writing $\phi_n^{(k)} = \hat{\theta}_n^{(k)} - \bar{\theta}_n^{(k)}$, we have

$$\phi_n^{(k)} = \begin{cases} \mathbf{f}(\mathbf{X}_1^{(k)}) - \bar{\theta}_k^{(k)} & \text{for } n = 1 \\ \phi_{n-1}^{(k)} - a_n \mathbf{Z}_n^{(k)} & \text{for } n \geq 2 \end{cases} \tag{3.20}$$

where

$$\mathbf{Z}_n^{(k)} = \begin{cases} \phi_{n-1}^{(k)} - (\mathbf{f}(\mathbf{X}_n^{(k)}) - \bar{\theta}_k^{(k)}) & \text{if } \mathbf{X}_n^{(k)} \in G(\hat{\mu}_{n-1}^{(k)}, \lambda_n) \\ \mathbf{0} & \text{otherwise} \end{cases} \tag{3.21}$$

This is because it is possible to write

$$\bar{\theta}_n^{(k)} = (1 - a_n)\bar{\theta}_n^{(k)} + a_n \bar{\theta}_n^{(k)}$$

and $0 < a_n \leq 1$ by choice.

Thus the propositions can be shown to be true if it can be shown that under the conditions assumed therein,

(i) $\phi_n^{(k)}$ converges almost surely to $\mathbf{0}$ as $n \to \infty$

(ii) $\{\mathcal{E}[\| \phi_n^{(k)} \|^2]\}$ converges as $n \to \infty$.

To establish these, the lemmas 3.2 and 3.3 are applied directly, by showing that conditions (B1)-(B7) are true for $\phi_n^{(k)}$ as defined by equations 3.20 and 3.21.

The condition (B1) is true because of the assumption (CG1), and the condition (B2) is seen to be hold in view of equations 3.20 and 3.21.

The requirement (B3) is also satisfied, with

$$\mathbf{M}_n(\mathbf{x}) = p_n^{(k)}\mathbf{x} \ ,$$

as

$$\begin{aligned}
&\mathcal{E}\left[\mathbf{Z}_{n+1}^{(k)} \mid \phi_1^{(k)}, \phi_2^{(k)}, \ldots, \phi_n^{(k)}\right] \\
&= p_n^{(k)}\left[\phi_n^{(k)} - \mathcal{E}[\mathbf{f}(\mathbf{X}_n^{(k)})] + \bar{\theta}_n^{(k)}\right] \\
&\qquad \text{since } \mathbf{X}_{n+1}^{(k)} \text{ is independent of } \mathbf{X}_1^{(k)}, \mathbf{X}_2^{(k)}, \ldots, \mathbf{X}_n^{(k)} \\
&\qquad \text{and hence of } \phi_1^{(k)}, \phi_2^{(k)}, \ldots, \phi_n^{(k)} \\
&= p_n^{(k)}\left[\phi_n^{(k)} - \mathcal{E}[\mathbf{f}(\mathbf{X}) \mid \hat{\omega} = k] + \bar{\theta}_n^{(k)}\right] \\
&= p_n^{(k)}\phi_n^{(k)}
\end{aligned}$$

as by the lemma 3.1,

$$\mathcal{E}[\mathbf{f}(\mathbf{X}) \mid \hat{\omega} = k] = \sum_{j=1}^{n} \beta_{kj}(n)\theta^{(j)} = \bar{\theta}_n^{(k)}. \tag{3.22}$$

Similarly, we have

$$\begin{aligned}
&\mathcal{E}\left[\| \mathbf{Z}_{n+1}^{(k)} \|^2 \mid \phi_1^{(k)}, \phi_2^{(k)}, \ldots, \phi_n^{(k)}\right] \\
&= p_n^{(k)}\mathcal{E}\left[\| \mathbf{Z}_{n+1}^{(k)} \|^2\right] \text{ by the condition (A5)} \\
&= p_n^{(k)}\left[\| \phi_n^{(k)} \|^2 + \mathcal{E}\left[\| \mathbf{f}(\mathbf{X}) - \bar{\theta}_n^{(k)} \|^2 \mid \hat{\omega} = k\right]\right] \text{ by the equation 3.22} \\
&\leq p_n^{(k)}\left[\| \psi_n^{(k)} \|^2 + \mathcal{E}\left[\| \mathbf{f}(\mathbf{X}) \|^2 \mid \hat{\omega} = k\right]\right] \\
&\qquad \text{as } \mathcal{E}\left[\| \mathbf{f}(\mathbf{X}) \|^2 \mid \hat{\omega} = k\right] \geq \mathcal{E}\left[\| \mathbf{f}(\mathbf{X}) - \bar{\bar{\theta}}^{(k)} \|^2 \mid \hat{\omega} = k\right]
\end{aligned}$$

$$= \quad p_n{}^{(k)} \left[\| \phi_n{}^{(k)} \|^2 + \sum_{j=1}^{m} \beta_{kj}(n)\rho_j \right] \text{ by the condition (CG2) and equation 3.5}$$

$$\leq \quad p_n{}^{(k)} \left[\| \phi_n{}^{(k)} \|^2 + \sum_{j=1}^{m} \rho_j \right] \quad \text{as } \beta_{kj}(n) \leq 1 \quad \forall k, j$$

Thus the requirement (B4) is seen to be satisfied with

$$a = \sum_{j=1}^{m} \rho_j, \quad b = 0, \quad c = 1.$$

The requirement (B5) is seen to be met as

$$\mathbf{x}^T \mathbf{M}_n(\mathbf{x}) = p_n{}^{(k)} \mathbf{x}^T \mathbf{x} \geq 0.$$

The condition (B6)is satisfied , because

$$\mathcal{E}\left[\| \phi_1{}^{(k)} \|^2 \right]$$

$$= \quad p_n{}^{(k)} \left[\mathcal{E}\left[\| \mathbf{f}(\mathbf{X}) - \bar{\theta}_n{}^{(k)} \|^2 \mid \hat{\omega} = k \right] \right]$$

$$\leq \quad p_n{}^{(k)} \left[\mathcal{E}\left[\| \mathbf{f}(\mathbf{X}) \|^2 \mid \hat{\omega} = k \right] \right]$$

$$= \quad p_n{}^{(k)} \left[\sum_{j=1}^{m} \beta_{kj}(n)\rho_j \right]$$

$$< \quad \infty$$

Finally, (B7) follows because

$$\inf_{\eta < \|\mathbf{x}\| < \eta^{-1}} \mathbf{x}^T \mathbf{M}_n \mathbf{x}$$

$$= \quad \inf_{\eta < \|\mathbf{x}\| < \eta^{-1}} p_n{}^{(k)} \mathbf{x}^T \mathbf{x}$$

$$> \quad p_n{}^{(k)} \eta^2$$

$$> \quad 0 \text{ by the condition (CG2)}$$

Hence the proposition.

The following result is an immediate consequence of the above proposition.

Proposition 3.5 *For the problem of estimating $\theta^{(k)}$ recursively under the setup defined in section 2.2, by the assumptions (A1)- (A6), let $\hat{\hat{\theta}}_n^{(k)}$ be the sequence of estimates used, where $\hat{\theta}_n^{(k)}$ is given by the equation 2.3. If there is a possibility of training samples being mislabeled, and the model assumed in section 3.2 is taken to be valid, and if*

(CG1) $\sum_{n=1}^{\infty} {a_n}^2 < \infty$,

(CG2) ${p_n}^{(k)} = P(A_k(n)|\hat{\omega} = k)$

$$= P\left(d_n(\hat{\mu}\,{}^{(k)}_{n-1}, \mathbf{X}_n{}^{(k)}) \leq \lambda_n \mid \hat{\mu}\,{}^{(k)}_{n-1}, \hat{\omega} = k\right) > \delta \quad \forall n \text{ for some } \delta > 0$$

and

(CG3) the statistic $\mathbf{f(X)}$ *admits of second-order moments, with*

$$\mathcal{E}\left[\| \mathbf{f(X)} \|^2 \mid \omega = i\right] = \rho_i$$

then

$$\sum_{j=1}^{m} \gamma_{kj}(n) \hat{\theta}\,{}^{(k)}_{n} \xrightarrow{a.s.} \theta^{(k)}$$

where

$$\Upsilon(n) = ((\upsilon_{ij}(n)))$$

is a generalized inverse [82] of the matrix

$$\mathrm{B} = ((\beta_{ij}(n)))$$

satisfying

$$\mathrm{B}(n)\Upsilon(n) = \mathbf{I}_m. \tag{3.23}$$

Proof: Firstly, we note that the matrix B need not be full- rank. From proposition 3.4, it is known that if $\mathrm{B}(n)^T$ denotes the transpose of $\mathrm{B}(n)$, then

$$\left(\hat{\theta}\,{}^{(1)}_{n} \middle| \hat{\theta}\,{}^{(2)}_{n} \middle| \cdots \middle| \hat{\theta}\,{}^{(m)}_{n}\right) - \mathrm{B}(n)^T \left(\theta^{(1)} \middle| \theta^{(2)} \middle| \cdots \middle| \theta^{(m)}\right)$$

$$\xrightarrow{a.s.} (\mathbf{0}|\mathbf{0}|\cdots|\mathbf{0}) \text{ element-wise}$$

i.e., every *element* of the matrix on the left-hand side converges almost surely to the corresponding *element* on the right-hand side). By well-known results on almost sure convergence it follows that

$$\left(\hat{\theta}\,{}^{(1)}_{n} \middle| \hat{\theta}\,{}^{(2)}_{n} \middle| \cdots \middle| \hat{\theta}\,{}^{(m)}_{n}\right) - \mathrm{B}(n)^T \left(\theta^{(1)} \middle| \theta^{(2)} \middle| \cdots \middle| \theta^{(m)}\right)$$

$$\xrightarrow{a.s.} (\mathbf{0}|\mathbf{0}|\cdots|\mathbf{0}) \text{ column-wise}$$

i.e., every *column* of the matrix on the left-hand side converges a.s. to the corresponding column on the right-hand side), and hence

$$\Upsilon(n)^T \left(\hat{\theta}\,{}^{(1)}_{n} \middle| \hat{\theta}\,{}^{(2)}_{n} \middle| \cdots \middle| \hat{\theta}\,{}^{(m)}_{n}\right)$$

$$\xrightarrow{a.s.} \left(\theta^{(1)} \middle| \theta^{(2)} \middle| \cdots \middle| \theta^{(m)}\right) \text{ column-wise}$$

by the equation 3.23, so that ultimately,

$$\Gamma^T \left(\hat{\theta}\,{}^{(1)}_{n} \middle| \hat{\theta}\,{}^{(2)}_{n} \middle| \cdots \middle| \hat{\theta}\,{}^{(m)}_{n}\right)$$

$$\xrightarrow{a.s.} \left(\theta^{(1)} \middle| \theta^{(2)} \middle| \cdots \middle| \theta^{(m)}\right) \text{ element-wise.}$$

Hence the proposition.

Notes

- Proposition 3.1 follows from proposition 3.4 on substituting δ_{kj} for $\beta_{kj}(n)$ $\forall k, j = 1, 2, \ldots, m$ and $\forall n \geq 1$.
- Proposition 3.2 follows as a consequence of proposition 3.4 by putting $\lambda_n = \infty$, that is, by making $p_n^{(k)} = 1$ so that the condition (CG2) is automatically satisfied by the non-GGA.
- In the non-ideal situation where there is a possibility of training samples being mislabeled, it is observed that for both the GGA and the non-GGA, almost sure convergence occurs,albeit in different forms but not towards the true values. In order to effect a comparison between the two algorithms in respect of (strong) convergence, it would be logical, therefore, to study how the sequence $\{\hat{\theta}_n^{(k)}\}$ behaves with respect to the true value of $\theta^{(k)}$. More specifically, one may wish to know whether $\{\hat{\theta}_n^{(k)}\}$ manages at all to get 'closer' *eventually* to $\theta^{(k)}$ than $\{\hat{\hat{\theta}}_n^{(k)}\}$ does. The following proposition establishes that $\hat{\theta}_n^{(k)}$ does indeed approach $\theta^{(k)}$ 'closer' than $\hat{\hat{\theta}}_n^{(k)}$ does, provided certain conditions are satisfied.

Proposition 3.6 *If, in addition to the assumptions (A1)-(A8), (CNG1), (CNG2) and (CG1), (CG2) and (CG3) we also have, for some k,*

(CG4) $\beta_{kj}(n) \rightarrow \beta_{kj}$
for $j = 1, 2, \ldots, m$ *as* $n \rightarrow \infty$ *for some* $\beta_{kj} \in [0, 1]$.
(CG5) $(\bar{\theta}_n^{(k)} - \theta^{(k)})^T (\bar{\bar{\theta}}^{(k)} - \bar{\theta}_n^{(k)}) > 0$
where $\bar{\theta}^{(k)} = \sum_{j=1}^m \epsilon_{kj}\theta^{(j)}$ *and* $\bar{\theta}_n^{(k)} = \sum_{j=1}^m \beta_{kj}\theta^{(j)}$.

then

$$\|\hat{\theta}_n^{(k)} - \theta^{(k)}\| - \|\hat{\hat{\theta}}_n^{(k)} - \theta^{(k)}\| \stackrel{a.s.}{\rightarrow} G_k$$

where G_k *is some number* > 0.

Proof. Under the assumption (CG4), it follows from proposition 3.4 that

$$\hat{\theta}_n^{(k)} \stackrel{a.s.}{\rightarrow} \bar{\theta}^{(k)}$$

with

$$\bar{\theta}_n^{(k)} = \sum_{j=1}^m \beta_{kj}\theta^{(j)}.$$

This, together with the proposition 3.2, implies that

$$\hat{\theta}_n^{(k)} - \theta^{(k)} \stackrel{a.s.}{\rightarrow} \bar{\theta}_n^{(k)} - \theta^{(k)}$$

and

$$\hat{\hat{\theta}}_n^{(k)} - \theta^{(k)} \xrightarrow{a.s.} \bar{\bar{\theta}}^{(k)} - \theta^{(k)}.$$

Consequently,

$$\|\hat{\hat{\theta}}_n^{(k)} - \theta^{(k)}\| - \|\hat{\theta}_n^{(k)} - \theta^{(k)}\| \xrightarrow{a.s.} \|\bar{\bar{\theta}}^{(k)} - \theta^{(k)}\| - \|\bar{\theta}_n^{(k)} - \theta^{(k)}\|.$$

However,

$$\begin{aligned}
&\|\bar{\bar{\theta}}^{(k)} - \theta^{(k)}\|^2 - \|\bar{\theta}_n^{(k)} - \theta^{(k)}\|^2 \\
&= \|\bar{\bar{\theta}}^{(k)}\|^2 - \|\bar{\theta}^{(k)}\|^2 - 2\theta^{(k)T}(\bar{\bar{\theta}}^{(k)} - \bar{\theta}_n^{(k)}) \\
&= \|\bar{\bar{\theta}}^{(k)} - \bar{\theta}_n^{(k)}\|^2 + 2\bar{\theta}_n^{(k)T}(\bar{\bar{\theta}}^{(k)} - \bar{\theta}_n^{(k)}) - 2\theta^{(k)T}(\bar{\bar{\theta}}^{(k)} - \bar{\theta}_n^{(k)}) \\
&= \|\bar{\bar{\theta}}^{(k)} - \bar{\theta}_n^{(k)}\|^2 + 2(\bar{\theta}_n^{(k)} - \theta^{(k)})^T(\bar{\bar{\theta}}^{(k)} - \bar{\theta}_n^{(k)}) \\
&> 0 \text{ because of (CG5)}
\end{aligned}$$

Hence the proposition.

3.6.1 Remarks

- This theorem formalizes some sufficient conditions under which the GGA provides estimates which are asymptotically "closer" to the respective true values than the usual non-GGA estimates.

- One implication of the condition (CG5) is that the proposition 3.6 will also be true if

  $$\bar{\bar{\theta}}^{(k)} \succ \bar{\theta}_n^{(k)} \succ \theta^{(k)}$$

  or if

  $$\theta^{(k)} \succ \bar{\theta}_n^{(k)} \succ \bar{\bar{\theta}}^{(k)},$$

  where the partial order relation '$\succ$' is defined as follows :

  for $\mathbf{a}, \mathbf{b} \in \mathcal{R}^N$, $\mathbf{a} \succ \mathbf{b}$ if $a_i > b_i$ for all $i = 1, 2, \ldots, N$.

  Generally speaking, these conditions signify that the proposition will be true only for those learning situations in which the configuration of the m classes is such that, for any given class, either

  a) the true mean $\theta^{(k)}$ is an interior point of the lower quantant of $\bar{\theta}_n^{(k)}$ which, in turn, is an interior point of the lower quantant of $\bar{\bar{\theta}}^{(k)}$,

  or

  b) the inclusion relations are true in the reverse order.

Obviously, then, whether or not GGA-estimates are asymptotically 'closer' to the true mean than the non-GGA estimates, is partly dependent on the nature of the problem, more specifically, the relative configurations of the different classes.

(By the lower quantant of any point $\mathbf{y}_0$ in the N-dimensional space $\mathcal{R}^N$, we mean the region

$$Q_L(\mathbf{y}_0) = \{\mathbf{y} : y_i \leq y_{i0}\ \forall i = 1, 2, \ldots, N\}.)$$

3.7 Dynamic behavior of the guard zone [8]

It is obvious from the previous discussion that, the choice of λ_n(the dimension of the guard zone) plays a crucial role so far as the convergence and classification efficiency of the GGA-estimates of the parameters are concerned. While it is not a very simple problem to obtain some sort of an optimal value without making additional assumptions, one can obtain certain bounds for λ_n from the view-point of convergence of the class parameters. The size of the guard zone may then be experimentally determined using some linear combination of those bounds.

As seen in section 3.6, one of the conditions necessary for having some form of stochastic convergence of the estimates to the true value is (CG2), that is,

$$\begin{aligned} p_n{}^{(k)} &= P(A_k(n)|\hat{\omega} = k) \\ &= P\left(d_n(\hat{\mu}_{n-1}^{(k)}, \mathbf{X}_n{}^{(k)}) \leq \lambda_n \mid \hat{\mu}_{n-1}^{(k)}, \hat{\omega} = k\right) \\ &> \delta \quad \forall n \text{ for some } \delta > 0 \end{aligned}$$

which requires that the probability of $d_n(\hat{\mu}_{n-1}^{(k)}, \mathbf{X}_n{}^{(k)})$ being less than or equal to the dimension of guard zone is strictly greater than zero.

By virtue of the lemma 3.4 given below it follows that

$$d_n{}^2(\hat{\mu}_{n-1}^{(k)}, \mathbf{X}_n{}^{(k)}) \geq \pi_{(n)min} \|\hat{\mu}_{n-1}^{(k)} - \mathbf{X}_n{}^{(k)}\|^2 = \ell_n{}^2, \text{ say,} \tag{3.24}$$

$$d_n{}^2(\hat{\mu}_{n-1}^{(k)}, \mathbf{X}_n{}^{(k)}) \leq \pi_{(n)max} \|\hat{\mu}_{n-1}^{(k)} - \mathbf{X}_n{}^{(k)})\|^2 = L_n{}^2, \text{ say.} \tag{3.25}$$

where $\pi_{(n)min}$ and $\pi_{(n)max}$ are respectively the smallest and largest eigenvalues of $\mathbf{A}_n$. As the $\mathbf{A}_n$'s are assumed to be symmetric positive definite, we must have

$$\pi_{(n)max} \geq \pi_{(n)min} \geq 0\ \forall n$$

that is,

$$0 \leq \ell_n \leq L_n\ \forall n.$$

Now, λ_n cannot be less than or equal to ℓ_n as that would mean that $p_n{}^{(k)} = 0$ which violates the condition (CG2). Also, $\lambda_n{}^{(k)}$ cannot be greater than or equal to L_n, as that would mean that $p_n{}^{(k)} = 1$, which is not desirable because all the samples would then be accepted by the supervisor for updating $\hat{\theta}_n^{(k)}$, the nth-stage estimate of $\theta^{(k)}$. Thus, one must necessarily have

$$\ell_n \leq \lambda_n{}^{(k)} \leq L_n\ \forall n. \tag{3.26}$$

The value of ${\lambda_n}^{(k)}$ is therefore found to be bounded between ℓ_n and L_n in order to have convergence of the estimates of classification parameters to their true values. From the relation 3.26 it is also interesting to note that the dimension of the guard zone is dynamic (varying) and its value at the nth stage depends on the $(n-1)$th stage-estimate of mean vector and the values of ${\ell_n}^{(k)}$ and ${L_n}^{(k)}$ i.e., the $(n-1)$th stage estimate of the matrix $\mathbf{A}_n{}^{(k)}$. This adaptive (expanding-shrinking) behavior of the guard zone $G(\mathbf{a}, {\lambda_n}^{(k)})$ centred at a enables the algorithm to accept sometimes a sample having a larger distance from a while discarding another one with a smaller distance for the parameter-updating. This was not the case with the experiments of Chien[54] and Pal et al. [53] where such a parameter was considered to be fixed throughout the learning process. It must be noted, however, that the former has in theory a provision for a dynamic threshold, though this has not been implemented there. In other words, the supervisory program uses here knowledge of its past behaviour, which depends on the input sequence.

In view of the condition 3.26 on the lower and upper bounds, we may take the following weighted average

$$\lambda_n{}^{(k)} = (1-\alpha){\ell_n}^{(k)} + \alpha {L_n}^{(k)},\ 0 < \alpha < 1, \tag{3.27}$$

of ${\ell_n}^{(k)}$ and ${L_n}^{(k)}$ in order to describe the dynamic behavior of the extent of the guard zone at the nth stage of learning. It is to be noted here that condition 3.26 is violated in case the matrix $\mathbf{A}_n{}^{(k)}$ is a scalar multiple of the identity matrix for any n.

Lemma 3.4 *If* $\mathbf{A}$ *is a symmetric matrix of order* p*, then*

$$\pi_p \leq \frac{\mathbf{x}^T\mathbf{A}\mathbf{x}}{\mathbf{x}^T\mathbf{x}} \leq \pi_1,\ \mathbf{x} \in \mathcal{R}^P,$$

where π_p *and* π_1 *are respectively the smallest and largest of the* p *roots of the equation*

$$|\mathbf{A} - \pi\mathbf{I}| = 0,$$

that is, they are respectively the smallest and largest eigenvalues of $\mathbf{A}$ *(both non-negative).* $\mathbf{I}$ *denotes the identity matrix of order* p*.*

3.7.1 A special case

In case we have

$$\left[\mathbf{A}_n{}^{(k)}\right]^{-1} = \begin{bmatrix} (\sigma_{1n}{}^{(k)})^2 & 0 & 0 & \dots & 0 \\ 0 & (\sigma_{2n}{}^{(k)})^2 & 0 & \dots & 0 \\ 0 & 0 & (\sigma_{3n}{}^{(k)})^2 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & (\sigma_{Nn}{}^{(k)})^2 \end{bmatrix}$$

where ${\sigma_{jn}}^2$ is the variance of the jth feature at the nth stage, then under the conditions 3.24 and 3.25,the bounds for ${\lambda_n}^{(k)}$ for the kth class will become

$${\ell_n}^{(k)} = \frac{\|\hat{\mu}_{n-1}^{(k)} - \mathbf{X}_n{}^{(k)}\|}{{\sigma_{(n-1)max}}^{(k)}}$$

and

$$L_n^{(k)} = \frac{\|\hat{\mu}_{n-1}^{(k)} - \mathbf{X}_n^{(k)}\|}{\sigma_{(n-1)min}^{(k)}}$$

where $\sigma_{(n-1)max}^{(k)}$ and $\sigma_{(n-1)min}^{(k)}$ denote respectively the largest and smallest values among the N standard deviation components in the kth class. From the above equations it is seen that if for a particular class we have further, for some n

$$\sigma_{1n}^{(k)} = \sigma_{2n}^{(k)} = \cdots = \sigma_{Nn}^{(k)}$$

then

$$\sigma_{(n)min}^{(k)} = \sigma_{(n)max}^{(k)}$$

and hence

$$\ell_n^{(k)} = L_n^{(k)} = \lambda_n^{(k)}.$$

In fact, the above equality will hold whenever $\mathbf{A}_n^{(k)}$ is a scalar multiple of the identity matrix.

Chapter 4

Automatic selection of thresholds for the GGA [9,10]

4.1 Introduction

It was observed in chapter 2 that the Generalised Guard-zone Algorithm (GGA) is an algorithm for learning class parameters for pattern recognition, which uses thresholds dynamically to implement a restricted updating program. Basically, its aim is to detect mislabeled training samples and outliers and to reject them from the parameter updating procedure. The algorithm is a generalisation of some existing ones [54,53] which were found to be useful for some sets of real-life data. In chapter 3, it was observed that the guard zone parameter $\lambda_n^{(k)}$ of the GGA lies between certain bounds and the recognition rate increases when the guard zone is made ***dynamic*** by causing its zone-controlling parameter $\lambda_n^{(k)}$ to be dependent on current estimates. It is to be mentioned here that this zone-controlling parameter was kept constant in the experiments of Pal et al.[53] and Chien [54], although the algorithm proposed by the latter does involve a dynamic threshold. The bounds were used to define certain estimates for $\lambda_n^{(k)}$ which gave good results (see chapter 5) *vis-à-vis* the non-GGA when applied to real-life data.

However, the problem of automatic selection of the guard-zone parameter $\lambda_n^{(k)}$ was not greatly facilitated by the above study. It continued to be an impediment in the practical implementation of the GGA. It became necessary therefore, to tackle this problem from a different view-point, that is, by using criteria other than stochastic convergence. This led to the results presented in this chapter, in which attempts have been made to determine automatically the values of guard zone dimension at every instant of learning on the basis of the criterion of minimum mean squared error (MSE). The explicit expressions for the mean squared error are obtained for both the GGA and the non-GGA (i.e., the usual unsupervised stochastic approximation learning algorithm not based on guard- zone) using the model of Chittineni [77] involving mislabeled training samples. An approximation to the guard-zone parameter is obtained for which the MSE for the GGA is smaller than that for the non-GGA. In other words, the value of $\lambda_n^{(k)}$ selected automatically by the system makes the GGA discard with greater efficiency the doubtful (mislabeled) samples from the training set, thus improving its performance *vis-à-vis* the non-GGA, for self-supervised learning.

4.2 Performance of the GGA relative to that of the non-GGA

Before a comparison of the performances of the two algorithms can be made, it is necessary to introduce a suitable measure of the quality of learning. Ideally, such a function should estimate at each instant n, the distance between the current state $\hat{\theta}_n$ and the optimal state θ. One convenient performance index of learning is the **mean squared error** of the estimate at each instant, namely,

$$D(\hat{\theta}_n) = \mathcal{E}\left[\|\hat{\theta}_n - \theta\|^2\right] \tag{4.1}$$

for discrete algorithms of learning.

In the following sections, we shall be dropping the suffix k denoting the class whenever possible, for convenience and brevity, unless required to do so in order to resolve ambiguity and confusion.

4.2.1 Performance index of the GGA

The GGA is defined as

$$\hat{\theta}_n^{(k)} = \begin{cases} \mathbf{f}(\mathbf{X}_1^{(k)}) & \text{for } n = 1 \\ \hat{\theta}_{n-1}^{(k)} - a_n \mathbf{Y}_n^{(k)} & \text{for } n \geq 2 \end{cases}$$

where

$$\mathbf{Y}_n^{(k)} = \begin{cases} \hat{\theta}_{n-1}^{(k)} - \mathbf{f}(\mathbf{X}_n^{(k)}) & \text{if } \mathbf{X}_n^{(k)} \in G(\hat{\mu}_{n-1}^{(k)}, \lambda_n) \\ \mathbf{0} & \text{otherwise} \end{cases}$$

and all symbols are as in section 2.2. Let

$$p_n^{(k)} = P(A_k(n)),$$

where

$$A_k(n) = \{\mathbf{X} : \mathbf{X} \in G(\hat{\mu}_{n-1}^{(k)}, \lambda_n^{(k)})\}.$$

Also,let

$$\bar{a}_n = 1 - a_n \ \forall n$$

and

$$q_n = 1 - p_n \ \forall n.$$

Then

$$\begin{aligned} \mathcal{E}(\hat{\theta}_n) &= p_n \mathcal{E}(\bar{a}_n \hat{\theta}_{n-1} + a_n \mathbf{f}(\mathbf{X}_n)) + q_n \mathcal{E}(\hat{\theta}_{n-1}) \\ &= (\bar{a}_n p_n + q_n)\mathcal{E}(\hat{\theta}_{n-1}) + a_n p_n \mathcal{E}(\mathbf{f}(\mathbf{X}_n)) \\ &= (\bar{a}_n p_n + q_n)[(\bar{a}_{n-1} p_{n-1} + q_{n-1})\mathcal{E}(\hat{\theta}_{n-2}) \\ &\quad + a_{n-1} p_{n-1} \mathcal{E}(\mathbf{f}(\mathbf{X}_{n-1}))] + a_n p_n \mathcal{E}(\mathbf{f}(\mathbf{X}_n)) \end{aligned}$$

$$= \cdots$$

$$\vdots$$

$$= \prod_{i=2}^{n} (\bar{a}_i p_i + q_i) \mathcal{E}(\hat{\theta}_1)$$

$$+ \sum_{j=2}^{n} \left(\prod_{i=j+1}^{n} (\bar{a}_i p_i + q_i) \right) a_j p_j \mathcal{E}\mathbf{f}(\mathbf{X}_j)) \tag{4.2}$$

if we follow the convention that

$$\prod_{i=m}^{n} (\bar{a}_i p_i + q_i) = 1 \ \forall m > n.$$

Writing

$$A_{i,j} = \prod_{k=i}^{j} (\bar{a}_k p_k + q_k) = \prod_{k=i}^{j} (1 - a_k p_k)$$

we have, from the equation 4.2,

$$\mathcal{E}(\hat{\theta}_n) = \sum_{j=1}^{n} A_{j+1,n} a_j p_j \mathcal{E}(\mathbf{f}(\mathbf{X}_j)). \tag{4.3}$$

This is because

$$a_1 = 1 \tag{4.4}$$

and

$$p_1 = 1. \tag{4.5}$$

If we define

$$Z_n = \mathbf{P}_n{}^T \mathbf{P}_n = \| \hat{\theta}_n - \theta \|^2$$

where

$$\mathbf{P}_n = \hat{\theta}_n^{(k)} - \theta^{(k)},$$

that is,

$$\mathbf{P}_n = \begin{cases} \mathbf{f}(\mathbf{X}_1{}^{(k)}) - \theta^{(k)} & \text{for } n = 1 \\ \bar{a}_n \mathbf{P}_{n-1} + a_n(\mathbf{f}(\mathbf{X}_n{}^{(k)})) - \theta^{(k)} & \text{with probability } p_n, n > 1 \\ \mathbf{P}_{n-1} & \text{with probability } q_n, n > 1 \end{cases}$$

then

$$Z_n = \begin{cases} \| \mathbf{f}(\mathbf{X}_1{}^{(k)}) - \theta^{(k)} \|^2 & \text{for } n = 1 \\ \bar{a}_n^2 Z_{n-1} + T_n & \text{with probability } p_n, n > 1 \\ Z_{n-1} & \text{with probability } q_n, n > 1 \end{cases}$$

where

$$T_n = a_n{}^2 \mathbf{Q}_n{}^T \mathbf{Q}_n + 2 a_n \bar{a}_n \mathbf{P}_{n-1}{}^T \mathbf{Q}_n$$

with

$$\mathbf{Q}_n = \mathbf{f}(\mathbf{X}_n) - \theta \ .$$

Thus $Z_1 = \mathbf{Q}_1{}^T\mathbf{Q}_1$ and

$$\mathcal{E}(Z_n) = \sum_{j=1}^{n} B_{j+1,n} p_j \mathcal{E}(T_j).$$

where

$$B_{i,j} = \prod_{k=i}^{j} (\bar{a}_k^2 p_k + q_k)$$

in view of the equation 4.5. As

$$\mathcal{E}(T_j) = a_j{}^2 \mathcal{E}(\mathbf{Q}_j{}^T\mathbf{Q}_j) + 2a_j\bar{a}_j\mathcal{E}(\mathbf{P}_{j-1}{}^T\mathbf{Q}_j)$$

and

$$\mathcal{E}(\mathbf{P}_{j-1}{}^T\mathbf{Q}_j) = 0 \ \forall j$$

on account of our assumption (A5) regarding the independence of the observations, we have

$$\mathcal{E}(T_j) = a_j{}^2 \mathcal{E}(\mathbf{Q}_j{}^T\mathbf{Q}_j).$$

Thus

$$\mathcal{E}(Z_n) = \sum_{j=2}^{n} B_{j+1,n} a_j{}^2 p_j E_{jj}{}^{(k)} \tag{4.6}$$

in view of the equations 4.4 and 4.5, where

$$E_{jj}{}^{(k)} = \mathcal{E}(\mathbf{Q}_j{}^T\mathbf{Q}_j | \hat{\omega} \ = k).$$

Let us now assume that the condition (CG3) holds, so that

$$\tau_j = \mathcal{E}\left[\|\mathbf{f}(\mathbf{X}) - \theta^{(j)}\|^2 | \omega \ = j\right] \text{ exists,}$$

implying that

$$\begin{aligned} E_{jj}{}^{(k)} \ = \ & \sum_{i=1}^{m} [\beta_{ki}(j)(\tau_i + \|\theta^{(i)} - \theta^{(k)}\|^2) \\ & + \beta_{ki}{}^*(j)(\|\mathbf{f}(\mathbf{0}) - \theta^{(i)})\|^2 + \|\theta^{(i)} - \theta^{(k)}\|^2)] \end{aligned} \tag{4.7}$$

4.2.2 Performance index of the non-GGA

The non-GGA is defined by the equation 2.5 as

$$\hat{\hat{\theta}}_n^{(k)} = \begin{cases} \mathbf{f}(\mathbf{X}_1{}^{(k)}) & \text{for } n = 1 \\ \hat{\hat{\theta}}_{n-1}^{(k)} - a_n \tilde{\mathbf{Y}}_n^{(k)} & \text{for } n \geq 2 \end{cases}$$

where

$$\tilde{\mathbf{Y}}_n^{(k)} = \hat{\hat{\theta}}_{n-1}^{(k)} - \mathbf{f}(\mathbf{X}_n{}^{(k)}) \ .$$

Clearly, proceeding as before,

$$\mathcal{E}(\hat{\hat{\theta}}_n) = A^*_{2,n}\mathcal{E}(\hat{\hat{\theta}}_n) + \sum_{j=2}^{n} a_j A^*_{j+1,n}\mathcal{E}\mathbf{f}(\mathbf{X}_j) \tag{4.8}$$

if we follow the convention that

$$\prod_{i=m}^{n}(\bar{a}_i p_i + q_i) = 1 \ \forall m > n.$$

Writing

$$A^*_{i,j} = \prod_{k=i}^{j}\bar{a}_k = \prod_{k=i}^{j}(1 - a_k)$$

we have, from the equation 4.8

$$\mathcal{E}(\hat{\hat{\theta}}_n) = \sum_{i=1}^{n} A^*_{j+1,n} a_j \mathcal{E}(\mathbf{f}(\mathbf{X}_n)). \tag{4.9}$$

This is because of the equation 4.4.

If we define

$$Z_n^* = \mathbf{P}^*_n{}^T\mathbf{P}^*_n = \|\hat{\hat{\theta}}_n - \theta\|^2$$

where

$$\mathbf{P}^*_n = \hat{\hat{\theta}}_n^{(k)} - \theta^{(k)},$$

then

$$Z_n^* = \begin{cases} \|\mathbf{f}(\mathbf{X}_1{}^{(k)}) - \theta^{(k)})\|^2 & \text{for } n = 1 \\ \bar{a}_n^2 Z_{n-1}^* + T_n^* & \text{for } n > 1 \end{cases}$$

where

$$T_n^* = a_n{}^2\mathbf{Q}_n{}^T\mathbf{Q}_n + 2a_n\bar{a}_n\mathbf{P}^*_{n-1}{}^T\mathbf{Q}_n$$

with

$$\mathbf{Q}_n = \mathbf{f}(\mathbf{X}_n) - \theta\ .$$

Thus $Z_1^* = \mathbf{Q}_1{}^T\mathbf{Q}_1$ and

$$\mathcal{E}(Z_n^*) = \sum_{j=1}^{n} B^*_{j+1,n}\mathcal{E}(T_j^*).$$

where

$$B^*_{i,j} = \prod_{k=i}^{j}\bar{a}_k^2.$$

As

$$\mathcal{E}(T_j^*) = a_j{}^2\mathcal{E}(\mathbf{Q}_j{}^T\mathbf{Q}_j) + 2a_j\bar{a}_j\mathcal{E}(\mathbf{P}^*_{j-1}{}^T\mathbf{Q}_j)$$

and

$$\mathcal{E}(\mathbf{P}^*_{j-1}{}^T\mathbf{Q}_j) = 0 \ \forall j$$

on account of our assumption (A5) regarding the independence of the observations, we have

$$\mathcal{E}(T_j^*) = a_j{}^2\mathcal{E}(\mathbf{Q}_j{}^T\mathbf{Q}_j).$$

Thus

$$\mathcal{E}(Z_n{}^*) = \sum_{j=2}^{n} B^*{}_{j+1,n} a_j{}^2 E^*{}_{jj}{}^{(k)} \tag{4.10}$$

in view of the equation 4.4, where

$$E^*{}_{jj}{}^{(k)} = \mathcal{E}(\mathbf{Q}_j{}^T \mathbf{Q}_j | \hat{\omega} = k) \tag{4.11}$$

$$= \sum_{i=1}^{m} \epsilon_{ki}(\tau_i + \|\theta^{(i)} - \theta^{(k)}\|^2) \tag{4.12}$$

As

$$\epsilon_{ki} = \beta_{ki}(n) + \beta_{ki}{}^*(n) \ \forall n > 0,$$

we must have

$$E^*{}_{jj}{}^{(k)} - E_{jj}{}^{(k)} = \sum_{i=1}^{m} \beta_{ki}{}^*(j)(\tau_i + \|\mathbf{f}(0) - \theta^{(i)})\|^2).$$

If

(CP1) $f_i(\mathbf{X}) > f_i(0) \ \forall i = 1, 2, \ldots, q$

then

$$E^*{}_{jj}{}^{(k)} > E_{jj}{}^{(k)}.$$

4.2.3 A comparison of the two performance indices

From the above discussion it follows that

$$\mathcal{E}(Z_n) < \mathcal{E}(Z_n{}^*)$$

if and only if

$$B_{j+1,n} p_j E_{jj} < B^*{}_{j+1,n} E^*{}_{jj} \ \forall j = 1, 2, \ldots, n. \tag{4.13}$$

Let us examine this set of necessary and sufficient conditions closely. First of all, we note that as

$$0 \leq \bar{a}_j^2 \leq \bar{a}_j^2 p_j + q_j \leq 1 \ \forall j,$$

we must have, for all i, j,

$$B^*{}_{i,j} \leq B_{i,j}. \tag{4.14}$$

Also, it is sufficient to consider the case where $E_{jj} > 0$, since the condition 4.13 is always trivially true when $E_{jj} = 0$. Rewriting the inequality 4.13 as

$$p_j \leq \frac{B^*{}_{j+1,n}}{B_{j+1,n}} \cdot \frac{E^*{}_{jj}}{E_{jj}}, \ j = 1, 2, \ldots, n \tag{4.15}$$

where $E_{jj} > 0$ for all j, we have, for $j = n$,

$$p_n \leq E^*{}_{nn} / E_{nn}$$

which is, in effect, redundant, if assumption (CP1) holds, by which $E^*_{jj}/E_{jj} \geq 1$ necessarily. Let us write

$$R_{j,k} = B^*_{j,k}/B_{j,k}$$

and

$$e_j = E^*_{jj}/E_{jj}.$$

Then the inequality 4.15 can be rewritten as

$$p_n \leq R_{j+1,n} e_j.$$

However, as $R_{j,k}$ is monotonically non-increasing in k for fixed j, the above inequality is equivalent to

$$R_{2,j} p_j / e_j \leq \lim_{n \to \infty} R_{2,n} = R, \text{say}, \tag{4.16}$$

as the inequality 4.15 must hold for all $n > j$. Let us examine the infinite product

$$R = \prod_{k=2}^{\infty} \left[\bar{a}_k^2/(\bar{a}_k^2 p_k + q_k) \right] = \prod_{k=2}^{\infty} (1 - c_k), \text{say},$$

where

$$\begin{aligned} c_k &= (1 - \bar{a}_k^2) q_k / (\bar{a}_k^2 p_k + q_k) \\ &= d_k (1 - p_k)/(1 - d_k p_k) \end{aligned}$$

with

$$d_k = 1 - \bar{a}_k^2.$$

From standard results on infinite products [83], it follows that a necessary and sufficient condition for R to converge is that

$$\sum_{k=2}^{\infty} c_k < \infty.$$

At this stage, we state and prove the following lemma :

Lemma 4.1 *Let $\{x_k\}$ be a sequence of positive numbers such that*

$$x_k = b_k (1 - c_k)/(1 - b_k c_k)$$

where $b_k, c_k \in (0, 1)$.

Then

$$\sum_{k=1}^{\infty} x_k < \infty$$

if

$$b_k > b_{k+1}$$

and

$$c_k < c_{k+1}$$

for all $k = 1, 2, 3, \ldots$.

Proof. Let us write

$$g(b, c) = b(1 - c)/(1 - bc).$$

Then

$$\frac{\partial g}{\partial b} > 0$$

and

$$\frac{\partial g}{\partial c} < 0,$$

implying that if $b_k > b_{k+1}$ and $c_k < c_{k+1}$, then

$$g(b_k, c) > g(b_{k+1}, c) \text{ whatever } c \text{ may be}$$

and

$$g(b, c_k) > g(b, c_{k+1}) \text{ whatever } b \text{ may be,}$$

so that

$$g(b_k, c_k) > g(b_k, c_{k+1}) > g(b_{k+1}, c_{k+1}) \text{ whatever k may be,}$$

that is,

$$x_k > x_{k+1} \ \forall k.$$

Hence by the D'alembert ratio test for the convergence of any series of positive terms, the lemma follows.

This lemma provides sufficient conditions for R to converge. These sufficient conditions are :

$$d_k > d_{k+1}, \text{ that is, } a_k > a_{k+1} \ \forall k,$$

and

$$p_k < p_{k+1} \ \forall k.$$

Let us return to the condition 4.16. We now have some sufficient conditions for R to exist. Our next problem is to find its value, if possible. For this purpose, we make use of the following lemma :

Lemma 4.2 *Let us consider the infinite product*

$$\prod_{k=2}^{\infty} (1 - r_k), \quad r_k \in (0, 1],$$

where

$$r_k = b_k(1 - c_k)/(1 - b_k c_k),$$

such that

i) $b_k, c_k \in (0, 1)$,

ii) $\lim_{k \to \infty} b_k = 0$

iii) $b_k > b_{k+1}$ *for all* $k = 1, 2, \ldots$,

iv) $c_k < c_{k+1}$ *for all* $k = 1, 2, \ldots$

If for all k,

$$c_k = b_{k+1}/b_k$$

then

$$\prod_{k=2}^{\infty}(1-r_k) = (1-b_2).$$

Proof. The lemma is rather obvious, as

$$\prod_{k=2}^{\infty}(1-r_k) = \lim_{n\to\infty}\prod_{k=2}^{n}(1-r_k)$$

and

$$\begin{aligned}\prod_{k=2}^{n}(1-r_k) &= \frac{1-b_2}{1-b_2c_2}\cdot\frac{1-b_3}{1-b_3c_3}\cdot\cdots\cdot\frac{1-b_n}{1-b_nc_n}\\ &= \frac{1-b_2}{1-b_nc_n} \text{ as } c_k = b_{k+1}/c_k\\ &\to (1-b_2) \text{ as } n\to\infty,\end{aligned}$$

as

$$1 \geq 1-b_nc_n > 1-b_n \to 1 \text{ as } n\to\infty,$$

implying that

$$\lim_{n\to\infty}(1-b_nc_n) = 1.$$

This lemma can be used directly by us for the solution of the problem at hand, namely, to find conditions under which the sequences $\{a_n\}$ and $\{p_n\}$ satisfy the condition 4.16. We shall establish presently how and why this is possible.

On applying the lemma, we get

$$\begin{aligned} p_n &= d_{k+1}/d_k &&= (1-\bar{a}^2_{k+1})/1-\bar{a}^2_k\\ R &= 1-d_2 &&= 1-\bar{a}^2_2\\ R_{2,j} &= \tfrac{(1-d_2)}{(1-d_jp_j)} \end{aligned}$$

so that the condition 4.16 becomes equivalent to

$$p_j \leq e_j(1-d_jp_j).$$

Obviously, a sufficient condition for this to hold is, therefore,

$$d_{j+1} \leq d_je_j/(1+d_je_j).$$

All the major conclusions arrived at in this section can be formally stated as follows :

Proposition 4.1 *Let* $\{\hat{\theta}_n^{(k)}\}$ *and* $\{\hat{\hat{\theta}}_n^{(k)}\}$ *be sequences of estimates defined by equations 2.3 and 2.5 respectively. Let* $D(\cdot)$ *be as defined by the equation 4.1. If*

(CP1) $\mathbf{f} : \mathcal{R}^N \to \mathcal{R}^q$ *is such that* $f_i(\mathbf{X}) \geq f_i(\mathbf{0})$, $\forall i = 1, 2, \ldots, q$

(CP2) $\{a_n\}$ *is strictly decreasing monotonically*

(CP3) $a_n \to 0$ *as* $n \to \infty$

(CP4) $p_n = (1 - \bar{a}_{n+1}^2)/(1 - \bar{a}_n^2)$ *where* $\bar{a}_k = 1 - a_k$

(CP5) $\bar{a}_{n+1}^2 > [(1 - e_n) - e_n \bar{a}_n^2]$ *for* $n = 1, 2, \ldots$

where $e_j = E^*_{jj}/E_{jj}$, E^*_{jj} *and* E_{jj} *being as given by equations 4.12 and 4.7 respectively, then*

$$D(\hat{\theta}_n^{(k)}) \leq D(\hat{\hat{\theta}}_n^{(k)}).$$

It is interesting to note that if we take $a_n = 1/n$, then all the requirements (CP1)-(CP5) are satisfied.

4.3 An approximation to λ_n

In this section we shall show that it is possible to obtain certain approximations to the zone-controlling parameter λ_n, if one is prepared to make some assumptions, which are stated below :

(L1) For every $k = 1, 2, \ldots, m$, the distribution of $\mathbf{X}_n^{(k)}$, that is, the feature vector having the 'given' label k (as opposed to its true label), is N-variate normal with mean vector

$$\bar{\bar{\theta}}^{(k)} = \sum_{j=1}^{m} \epsilon_{kj} \theta^{(j)}$$

and dispersion matrix

$$\bar{\bar{\Sigma}}^{(k)} = \sum_{j=1}^{m} \epsilon_{kj} \Sigma^{(j)}$$

where

$$\epsilon_{kj} = \alpha_{kj}\pi_j / (\sum_{i=1}^{m} \alpha_{ki}\pi_i).$$

(L2) Let

$$\bar{\theta}_n^{(k)} = \sum_{j=1}^{m} \beta_{kj}(n) \theta^{(j)}.$$

and

$$\bar{\Sigma}_{nk}^{(k)} = \sum_{j=1}^{m} \beta_{kj}(n) \Sigma^{(j)}.$$

Then $\bar{\theta}_n^{(k)}$ can be approximated by $\bar{\bar{\theta}}^{(k)}$ and $\bar{\Sigma}_n^{(k)}$ by $\bar{\bar{\Sigma}}^{(k)}$.

Remark: One situation in which the conditions L1 and L2 can be surely expected to hold is in the ideal case, i.e., where there is no mislabeling. In such a situation these conditions become respectively equivalent to

(**L1′**) For every $k = 1, 2, \ldots, m$, the distribution of $\mathbf{X}_n^{(k)}$, i.e., the feature vector having the 'given' label k (as opposed to the true label), is N-variate normal with mean vector

$$\bar{\theta}^{(k)} = \theta^{(k)}$$

and dispersion matrix

$$\bar{\bar{\Sigma}}^{(k)} = \Sigma^{(k)}.$$

(**L2′**) $\bar{\theta}^{(k)}$ is equal to $\bar{\theta}_n^{(k)}$, and $\bar{\bar{\Sigma}}^{(k)}$ to $\bar{\bar{\Sigma}}_n^{(k)}$.

Thus whenever we assume $L1'$ and $L2'$ to hold, we are, in effect, assuming the absence of mislabeling in the training set, an assumption which may not always be justified. So, any results based on them will, at best, yield approximate results. However, they had to be resorted to in order to make the problem and its treatment tractable enough to yield useful results. We have the following result :

Proposition 4.2 *Let* $\{\hat{\theta}_n^{(k)}\}$ *and* $\{\hat{\hat{\theta}}_n^{(k)}\}$ *be sequences of estimates defined by equations 2.3 and 2.5 respectively, and let the assumptions (A1)-(A8), (CN1), (CN2), (CG1)-(CG3), (L1) and (L2) hold, under the setup assumed in section 3.2. Also, let*

$$a_n = 1/n,$$

$$\mathbf{A}_n^{-1} = \frac{1}{(n-1)} \sum_{j=1}^{n-1} (\mathbf{X}_j - \hat{\mu}_{n-1}^{(k)})(\mathbf{X}_j - \hat{\mu}_{n-1}^{(k)})^T.$$

Then for $n > N$, *a large-sample approximation to* λ_n, *which minimizes the mean-squared-error (MSE) of the GGA is given by*

$$\lambda_n^2 = \frac{n(n-1)}{(n-2)} u_{p_n}/(1 - u_{p_n}), \tag{4.17}$$

where u_p *is the lower p-percentage point of the beta distribution with degrees of freedom* $N/2$ *and* $(n-N)/2$, *so that*

$$p = \frac{1}{\mathrm{B}(\frac{N}{2}, \frac{(n-N)}{2})} \int_0^{u_p} u^{(N/2)-1}(1-u)^{((n-N)/2-1)} du,$$

with

$$\mathrm{B}(\frac{N}{2}, \frac{(n-N)}{2}) = \int_0^1 u^{(N/2)-1}(1-u)^{((n-N)/2-1)} du,$$

and

$$p_n = p_n^{(k)} = P(\{\mathbf{X} : \mathbf{X} \in G(\hat{\mu}_{n-1}^{(k)}, \lambda_n^{(k)})\})$$

is prespecified for all values of n.

Proof. Since the assumptions (A1)-(A8), (CN1), (CN2), (CG1)-(CG2) hold, it follows from the propositions 3.2 and 3.4 respectively that

$$\hat{\hat{\theta}}_n^{(k)} \xrightarrow{a.s.} \bar{\theta}^{(k)}$$

and

$$\hat{\theta}_n^{(k)} \xrightarrow{a.s.} \bar{\theta}_n^{(k)},$$

implying that

$$(\hat{\hat{\theta}}_n^{(k)} - \bar{\theta}^{(k)}) - (\hat{\theta}_n^{(k)} - \bar{\theta}_n^{(k)}) \xrightarrow{a.s.} 0. \tag{4.18}$$

By the model assumed in section 3.2, we have

$$\mathbf{X}_n{}^{(k)} \sim \mathcal{N}_N(\bar{\theta}^{(k)}, \bar{\bar{\Sigma}}^{(k)})$$

under (L2), where the notation $\sim$ denotes *is distributed as* and $\mathcal{N}_N(\mu, \Sigma)$ is the N-variate normal or Gaussian variable with mean vector μ and dispersion matrix Σ.

Also, as $a_n = 1/n$, it follows that $\hat{\hat{\theta}}_n^{(k)}$ is nothing but the *arithmetic mean* of the n observations $\mathbf{X}_1{}^{(k)}, \mathbf{X}_2{}^{(k)}, \ldots, \mathbf{X}_n{}^{(k)}$, so that from well-known results of statistical sampling theory [60], it follows, on account of (L1), that

$$\hat{\hat{\theta}}_n^{(k)} \sim \mathcal{N}_N(\bar{\theta}^{(k)}, \frac{1}{n}\bar{\bar{\Sigma}}^{(k)}). \tag{4.19}$$

The relations 4.18 and 4.19 together imply that

$$\sqrt{n}(\hat{\theta}_n^{(k)} - \bar{\theta}_n^{(k)}) \xrightarrow{\mathcal{L}} \mathcal{N}_N(0, \bar{\bar{\Sigma}}_k^{(k)}), \tag{4.20}$$

where the notation $\xrightarrow{\mathcal{L}}$ denotes 'convergence in distribution or $\mathcal{L}$ aw'.

By (A5), $\mathbf{X}_{n+1}{}^{(k)}$ is independent of $\mathbf{X}_1{}^{(k)}, \mathbf{X}_2{}^{(k)}, \ldots, \mathbf{X}_{n-1}{}^{(k)}$ and hence of

$$\sqrt{n}(\hat{\theta}_n^{(k)} - \bar{\theta}_n^{(k)})$$

so that

$$\frac{n}{n+1}(\mathbf{X}_{n+1}{}^{(k)} - (\hat{\theta}_n^{(k)} - \bar{\theta}_n^{(k)})) \xrightarrow{\mathcal{L}} \mathcal{N}_N(\bar{\theta}^{(k)}, \bar{\bar{\Sigma}}^{(k)}). \tag{4.21}$$

Also, for $i < n+1$,

$$\frac{n}{n+1}(\mathbf{X}_i{}^{(k)} - (\hat{\theta}_n^{(k)} - \bar{\theta}_n^{(k)})) \xrightarrow{\mathcal{L}} \mathcal{N}_N(\bar{\theta}^{(k)}, \bar{\bar{\Sigma}}^{(k)}) \tag{4.22}$$

as, by the equation 4.18,

$$\mathcal{E}[(\mathbf{X}_i{}^{(k)} - \hat{\theta}_n^{(k)})(\mathbf{X}_i{}^{(k)} - \hat{\theta}_n^{(k)})^T] - \mathcal{E}[(\mathbf{X}_i{}^{(k)} - \hat{\hat{\theta}}_n^{(k)})(\mathbf{X}_i{}^{(k)} - \hat{\hat{\theta}}_n^{(k)})^T]$$

$$\rightarrow \mathbf{0}_{N\times N},$$

where $\mathbf{0}_{N\times N}$ is the $N \times N$ matrix having all its elements equal to 0. Also, it can easily be observed that

$$\lim_{n\to 0} \mathcal{E}[(\mathbf{X}_i{}^{(k)} - \bar{\theta}^{(k)}) - (\hat{\theta}_n^{(k)} - \bar{\theta}_n^{(k)})][(\mathbf{X}_j{}^{(k)} - \bar{\theta}^{(k)}) - (\hat{\theta}_n^{(k)} - \bar{\theta}_n^{(k)})]^T$$

$$= \begin{cases} (1-\frac{1}{n})\bar{\bar{\Sigma}}^{(k)} & \text{if } i = j \\ -\frac{1}{n}\bar{\bar{\Sigma}}^{(k)} & \text{if } i \neq j \end{cases} \tag{4.23}$$

Applying (L2) to the relations 4.21 and 4.22 gives

$$\sqrt{n/(n+1)}(\mathbf{X}_{n+1}{}^{(k)} - \hat{\theta}_n^{(k)}) \xrightarrow{\mathcal{L}} \mathcal{N}_N(0, \bar{\bar{\Sigma}}^{(k)}) \tag{4.24}$$

and, for $i < n+1$,

$$\sqrt{n/(n-1)}(\mathbf{X}_i{}^{(k)} - \hat{\theta}_n^{(k)}) \xrightarrow{\mathcal{L}} \mathcal{N}_N(0, \bar{\bar{\Sigma}}^{(k)}) \tag{4.25}$$

The relation 4.23 implies that

$$\frac{n}{n-1}\sum_{i=1}^{n}[(\mathbf{X}_i{}^{(k)} - \bar{\theta}^{(k)}) - (\hat{\theta}_n^{(k)} - \bar{\theta}_n^{(k)})][(\mathbf{X}_i{}^{(k)} - \bar{\theta}^{(k)}) - (\hat{\theta}_n^{(k)} - \bar{\theta}_n^{(k)})]^T$$

$$\sim \mathcal{W}_N(\bar{\bar{\Sigma}}^{(k)}, n-1), \tag{4.26}$$

the N-variate central Wishart distribution with $n-1$ degrees of freedom, so that we are justified in claiming that

$$\frac{n}{n-1}\sum_{i=1}^{n}[\mathbf{X}_i{}^{(k)} - \hat{\theta}_n^{(k)}][\mathbf{X}_i{}^{(k)} - \hat{\theta}_n^{(k)}]^T$$

$$\xrightarrow{\mathcal{L}} \mathcal{W}_N(\bar{\bar{\Sigma}}^{(k)}, n-1), \tag{4.27}$$

that is,

$$\frac{n^2}{n-1}[\mathbf{A}_{n+1}]^{-1} \xrightarrow{\mathcal{L}} \mathcal{W}_N(\bar{\bar{\Sigma}}^{(k)}, n-1). \tag{4.28}$$

The relations 4.24 and 4.28 together imply that

$$\frac{(n-1)^2}{n(n+1)}(\mathbf{X}_{n+1}{}^{(k)} - \hat{\theta}_n^{(k)})\mathbf{A}_{n+1}(\mathbf{X}_{n+1}{}^{(k)} - \hat{\theta}_n^{(k)})^T$$

$$\xrightarrow{\mathcal{L}} \mathcal{T}^2{}_{n-1},$$

the N-variate central Hotelling-$\mathcal{T}^2$ distribution with $n-1$ degrees of freedom, that is,

$$\frac{(n-1)^2}{n(n+1)}d^2{}_{n+1}(\mathbf{X}_{n+1}{}^{(k)}, \hat{\mu}_n^{(k)}) \xrightarrow{\mathcal{L}} \mathcal{T}^2{}_{n-1}. \tag{4.29}$$

The proposition follows from this if we remember that

- If $\mathcal{T}^2$ is an N-variate central Hotelling $\mathcal{T}^2$-statistic with k degrees of freedom, then

$$\frac{k-N+1}{N}(\mathcal{T}^2/k) \sim \mathcal{F}(N, k-N+1),$$

the central $\mathcal{F}$-statistic with $(N, k-N+1)$ degrees of freedom.

- If $\mathcal{F}$ is a central $\mathcal{F}$-statistic with (m, n) d.f. then

$$U = \frac{c\mathcal{F}}{(1 + c\mathcal{F})} \sim Beta(m/2, n/2),$$

the Beta variate with $(m/2, n/2)$ d.f., where $c = m/n$.

We therefore have

$$\begin{aligned} p_{n+1} &= P\left[d^2{}_{n+1} \leq \lambda^2{}_{n+1}\right] \\ &= P\left[\frac{n(n+1)}{(n-1)^2} T^2{}_{n-1} \leq \lambda^2{}_{n+1}\right] \\ &= P\left[c(n-1)\mathcal{F} \leq \frac{(n-1)^2}{n(n+1)} \lambda^2{}_{n+1}\right] \\ &= P\left[\frac{U}{(1-U)} \leq \frac{(n-1)}{n(n+1)} \lambda^2{}_{n+1}\right] \end{aligned}$$

implying that

$$\frac{u_{p_n}}{(1 - u_{p_n})} = \frac{(n-1)}{n(n+1)} \lambda^2{}_{n+1}$$

This proves the proposition.

Remarks

1. Karl Pearson has tabulated the incomplete beta function

$$I_x(m, n) = \frac{1}{\mathrm{B}(m, n)} \int_0^x u^{m-1}(1-u)^{n-1} du$$

for a large number of values of m and n, in [84]. It is not difficult to determine the approximation to λ_n given above with the help of these tables.

2. Tables for the $\mathcal{F}$-statistic are also available, but they are not so extensive as those for the beta variate. Also, the probability density of the latter is more convenient to manipulate than that of the former, so that it is better to use the beta distribution even in case the percentage points are to be directly computed by the computer itself, either at the time of or before the algorithm is to be implemented.

3. These approximations to λ_n depend only on the dimension N of the feature vector, apart from n. So it is possible to tabulate their values for different N for a large number of values of n, for purposes of ready reference.

4. The point mentioned just above actually highlights a distinct advantage of the given method for estimating λ_n , as compared to the methods used earlier (see section 3.7). The latter involve a fair amount of computation, as the eigenvalues of an $N \times N$ matrix have to be computed at each iteration. Further, the values of λ_n have to be computed afresh for every new problem.

5. A word of caution is necessary here. The given method is only an approximate one and is based mostly on large-sample theory. So it is quite possible that the values obtained may not yield very satisfactory results in small-sample situations.

Chapter 5

Implementation and Experimental Results

5.1 Introduction

With a view to demonstrating the theoretical results regarding the GGA that were stated and proved in chapters 3 and 4, some experiments on real-life as well as simulated data, were undertaken. The GGA was implemented on

1) a simulated three-class two-feature pattern recognition experiment

2) a six-class three-feature Telugu vowel recognition problem

3) a five-class two-feature terrain classification problem based on LANDSAT-V data.

Details regarding these data sets are given in sections 5.2.1, 5.2.2 and 5.2.3 respectively.

Basically, the following experimental investigation was undertaken, assuming that a certain amount of mislabeling occurs inevitably :

- For different values of $\alpha \in (0,1)$, the values of guard-zone parameter ${\lambda_n}^{(k)}$ are computed according to the equation 3.33 in section 3.7, that is,

$$ {\lambda_n}^{(k)} = (1-\alpha){\ell_n}^{(k)} + \alpha {L_n}^{(k)} \tag{5.1} $$

  ${\ell_n}^{(k)}$ and ${L_n}^{(k)}$ being respectively the lower and the upper bounds for ${\lambda_n}^{(k)}$, derived in that section. For each such α, the sequence of ${\lambda_n}^{(k)}$-values so obtained is used for implementing the GGA to the data set. This sequence of λ-values will be called λ-*sequence* 1 for ease of reference.

- The sequence of λ_n-values, is computed with the help of equation 4.17 in chapter 4, that is,

$$ {\lambda_n}^2 = \frac{n(n-1)}{(n-2)} u_{p_n}/(1-u_{p_n}), \tag{5.2} $$

  where u_p is the lower p-percentage point of the beta distribution with degrees of freedom $N/2$ and $(n-N)/2$, so that

$$ p = \frac{1}{\mathrm{B}(\frac{N}{2}, \frac{(n-N)}{2})} \int_0^{u_p} u^{(N/2)-1}(1-u)^{((n-N)/2-1)} du, $$

with

$$\mathrm{B}(\frac{N}{2}, \frac{(n-N)}{2}) = \int_0^1 u^{(N/2)-1}(1-u)^{((n-N)/2-1)}du,$$

and

$$p_n = p_n{}^{(k)} = P(\{\mathbf{X} : \mathbf{X} \in G(\hat{\mu}_{n-1}^{(k)}, \lambda_n{}^{(k)})\})$$

is *prespecified for all values of* n,

This sequence of λ_n-values is used for implementing the GGA to the data set. This sequence will subsequently be referred to as λ-*sequence*2.

- The performance of the GGA in both cases is compared to that of the non-GGA (that is, the nonsupervised system (NS, in short)) and/or the fully-supervised system (or FS, in brief) on the basis of the percentage of correct recognition at every iteration. (The terms 'nonsupervised'and 'fully-supervised' are defined at the end of this section.)

- In addition, the result presented as proposition 3.6, which states that the GGA manages to get closer to the true parameter value in the long run than the non-GGA does, provided certain conditions are satisfied, is also verified for every implementation of the GGA.

As the estimate at any given iteration is dependent on the order in which the training samples appear in the input sequence, a number of different (random) orderings or permutations (i.e., 'sequences') of the training set is used.

In addition, the following were also assumed regarding the GGA, that is, the following were substituted in equation 2.3:

1) $a_n = 1/n \quad \forall n$,

2) the Mahalanobis distance [60] for the distance measure $d(\cdot,\cdot)$.

The Bayes classifier

For an m-class pattern recognition problem based on an N-variate feature vector $\mathbf{X}$, the Bayes classification rule is [38]

Classify $\mathbf{X}$ into the class C_k if and only if

$$\pi_k p_k(\mathbf{X}) > \pi_j p_j(\mathbf{X}) \quad \forall j \neq k,$$

where π_j is the *a priori* probability for class C_j and $p_j(\cdot)$ is the class-conditional probability density for the jth class.

If $\forall k, \quad p_k(\mathbf{X}) = \mathcal{N}_N(\mu_k, \Sigma_k)$, the N-variate Gaussian or normal distribution with mean vector μ and covariance matrix Σ, then the decision rule is,

decide $\mathbf{X} \in C_k$ if and only if

$$D_k(\mathbf{X}) = \max_{1 \le j \le m} D_j(\mathbf{X}),$$

where

$$D_j(\mathbf{X}) = \ln |\Sigma_j| (\mathbf{X} - \mu_j)^T \Sigma_j^{-1} (\mathbf{X} - \mu_j).$$

Whenever a Bayes classifier has been used in this chapter, as with the Telugu vowel and LANDSAT data sets, we have assumed equal *a priori* probabilities for all the classes, that is,

$$\pi_i = \frac{1}{m}, \quad i = 1, 2, \ldots, m.$$

The experimental procedure

For the simulated PR experiment the experimental procedure is slightly simpler than for the others, since the labels (whether correct or not) are prespecified (see sections 5.2.1 and 5.3.1), and not provided by a classifier. An initial estimate for the mean vector and the covariance matrix is obtained on the basis of a single sample. With this initial estimate, the GGA and the non-GGA are run on the data set for the given choice of the λ-sequence. Since the data sets are small, the use of the % recognition score as a basis of comparison, is not really meaningful. Instead, the distance of the estimates from the *true* values is used for this purpose. The results are given in section 5.3.1.

For the Telugu vowel and LANDSAT data sets, the procedure is slightly different. Initial estimates of the parameters of the classifier are obtained on the basis of a certain (prespecified) number of samples. These estimates are used to initialize the classifier, which is then used to classify the first sample in the input sequence. The label provided by the classifier for that sample is accepted without any modification, and the parameters are updated by the GGA. This, in turn, is used to modify the classifier and so the process goes on, that is, learning and recognition take place concurrently, the results of the latter being used as labels for samples that are subsequently used for learning. In other words, *self-supervision* is said to take place. This type of self-supervised learning is done with both λ-sequences, as also with a GGA with fixed λ-values. For purposes of further comparison, learning is also done with:

1. a *fully-supervised* scheme, where the decision of the classifier is verified by an external supervisor and the class parameters are updated only if the classification is found to be correct;

2. a *nonsupervised* scheme, that is, the non-GGA, which accepts the label provided by the classifier and updates the estimate at every iteration without doing any kind of supervision.

As both these data sets are large, comparisons have been done on the basis of the % recognition score as well as the distance from the true parameter value (which is nothing but the parameter value for the entire data set). The corresponding results are displayed graphically. In some cases, different sample sizes for obtaining the initial estimates are also used. Details are provided in section 5.3.2 for the Telugu vowel data set, and in section 5.3.3 for the LANDSAT data set.

In all the experiments, the parameters to be estimated are the mean vectors and the covariance matrices. However, it should be noted that the *uncorrected (raw)* second-order moments (that is, elements of the covariance matrix) have been estimated, rather than the central second-order moments. In other words, we have estimated $\mathcal{E}(\mathbf{X}\mathbf{X}^T)$ rather than $\mathcal{E}(\mathbf{X}-\mu\ )(\mathbf{X}-\mu\ )^T$. This had to be done as condition (CP1) of proposition 4.1 would not have been satisfied otherwise.

5.2 Details of the data sets used

5.2.1 The simulated pattern recognition experiment

A three-class two-feature pattern recognition problem is simulated as follows, assuming that the feature vector has a Gaussian distribution in each of the classes.

The data set was generated using random normal deviates from [85], with mean vectors and dispersion matrices as given in Table 5.1 for each of the three classes. The method used for obtaining a sample (X, Y) from a bivariate normal population $\mathcal{N}_2(\mu\ , \Sigma\ )$, where

$$\mu\ = \begin{bmatrix} \mu_X \\ \mu_Y \end{bmatrix}$$

and

$$\Sigma\ = \begin{bmatrix} \sigma_{XX} & \sigma_{XY} \\ \sigma_{XY} & \sigma_{YY} \end{bmatrix},$$

from a pair of random normal deviates (τ_X, τ_Y) is based on the following well-known results :

Lemma 5.1 *If (X, Y) is distributed in the bivariate normal form with mean vector*

$$\mu\ = \begin{bmatrix} \mu_X \\ \mu_Y \end{bmatrix}$$

and dispersion matrix

$$\Sigma\ = \begin{bmatrix} \sigma_{XX} & \sigma_{XY} \\ \sigma_{XY} & \sigma_{YY} \end{bmatrix},$$

where

$$\sigma_{XY} = \rho\sqrt{\sigma_{XX}\sigma_{YY}},$$

then $(Y|x)$ is also normally distributed with

$$\mathcal{E}(Y|x) = \mu_{\ Y|x} = \mu_Y + \rho\sqrt{\frac{\sigma_{YY}}{\sigma_{XX}}}(x - \mu_X)$$

and

$$var(Y|x) = {\sigma_{Y|x}}^2 = \sigma_{YY}(1 - \rho^2).$$

Lemma 5.2 *Given two random variables X and Y, if $\mathcal{E}(Y|x)$ and $var(Y|x)$ exist for almost all values of x, then*

$$\mathcal{E}(Y) = \mathcal{E}_X \mathcal{E}(Y|X)$$

and

$$var(Y) = \mathcal{E}_X var(Y|X) + \mathcal{E}_X [E(Y|X) - E(Y)]^2.$$

As any pair (τ_X, τ_Y) of random normal deviate is equivalent to two *independent* normal deviates, if we take

$$X = \mu_X + \tau_X \sqrt{\sigma_{XX}}$$

and

$$Y = \mu_{Y|x} + \tau_Y \sigma_{Y|x},$$

then it is not difficult to verify that

$$\begin{bmatrix} X \\ Y \end{bmatrix} \sim \mathcal{N}_2 \left(\begin{bmatrix} \mu_X \\ \mu_Y \end{bmatrix}, \begin{bmatrix} \sigma_{XX} & \sigma_{XY} \\ \sigma_{XY} & \sigma_{YY} \end{bmatrix} \right),$$

with

$$\sigma_{XY}{}^2 = \rho^2 \sigma_{XX} \sigma_{YY}.$$

From the sets of samples so obtained for each of the three classes, training sets of size 20 for each were obtained by mixing at random the elements of the three sample sets, using the following $((\alpha_{ij}))$-matrix :

$$((\alpha_{ij})) = \frac{1}{20} \begin{bmatrix} 17 & 1 & 2 \\ 1 & 16 & 3 \\ 2 & 3 & 15 \end{bmatrix}.$$

This means, for example, that the training set for class 1 contains 17 samples from class 1, 1 sample from class 2 and so on. Two such data sets were constructed. They have been called Artificial Data Sets I and II (ADS-I and ADS-II, in short) for ease of reference. The performance of the GGA with *λ-sequence* 1 has been examined experimentally with the help of ADS-I. Table 5.2.1 gives the values of the various parameters related to the data set ADS-I, namely, $\mu_k, \Sigma_k, \bar{\bar{\theta}}^{(k)}$ and $\bar{\bar{\Sigma}}^{(k)}$, $k = 1, 2, 3$. Details regarding the data set ADS-II are given in Table 5.11.

5.2.2 The Telugu vowel data set

In order to obtain a data set based on the first three formant frequencies (F1,F2 and F3) of Telugu vowels, in the CNC (Consonant-Vowel Nucleus-Consonant) context the following procedure was used [53] :

A vocabulary consisting of Telugu words was selected so as to include as many CN and NC combinations as possible, with an emphasis on the use of commonly-used words. These were recorded by five adult male speakers (in the age-group of 30-35 years) on an AKAI tape recorder in a large auditorium. On the basis of a listening experiment by 10 listeners, only three speakers, denoted X,Y,Z were selected. A spectrographic

Class k	Mean Vector μ_k	Covariance Matrix Σ_k		Modified Mean Vector $\bar{\theta}^{(k)}$	Modified Covariance Matrix $\bar{\Sigma}^{(k)}$	
1	(10, 15)'	103	152	(9.25, 14.00)'	92.00	135.35
		152	233		135.35	209.80
2	([illegible], 5)'	29	25	(5.25, 6.25)'	32.85	34.95
		25	29		34.95	50.30
3	(5, 10)'	30	49	(5.50, 9.75)'	37.15	55.70
		49	103		55.70	104.90

Table 5.1: Some parameter values related to the Artificial Data Set I

analysis of these utterances was done on a Kay sonagraph (model 7029A). The analyses were carried out in the normal mode, using the band 80 Hz–8 KHz with wide band filters having bandwidth 300 Hz.

Formant frequencies $F1$, $F2$ and $F3$ were obtained manually at the steady state of the vowels. In view of the large amount of data to be handled, the formant frequencies were measured from the base line with a specially constructed scale. Reckecking on 5% of the samples revealed that the formant frequencies had been recorded within a accuracy of 10 Hz. Occasionally, steady states were not observed due to the extreme shortness of the vowels. Here measurements were made at the point of congruence of the off-glide and the on-glide. The samples which did not depict a prominent third formant were allowed to have an injected average third formant $(F3)_{av}$, computed over all members of that class of vowels for that particular speaker. The number of samples which fell in this category was 384. The measured values of the three features $F1$, $F2$ and $F3$ were therefore thought to constitute a three-dimensional feature space. Further details of the extraction procedure are available in [86,87]. Altogether 871 samples were collected. The distribution of these samples in the $(F_1 - F_2)$–plane is shown in figure 5.1.

There are ten vowel classes ($\partial:, a:, i, i:, e, e:, u, u:, o$ and $o:$) including long and short categories. Since the short and long categories of a vowel differ only in duration, these were pooled together resulting in six groups ($\partial, a:, I, E, U$ and O) which differ only in phonetic features. Incidentally, although the shorter and longer types of vowels I, E, U and O are treated similarly, they were given individual class parameter values. The set of data for each class has been found to follow the normal distribution [86,87]. Therefore, the use of the Bayes classifier for normally distributed patterns and the assumption made in chapter 2 that 'the probability of misclassification of the input patterns falling within the guard zone constructed around the central tendency of a class distribution is substantially low' are well justified here.

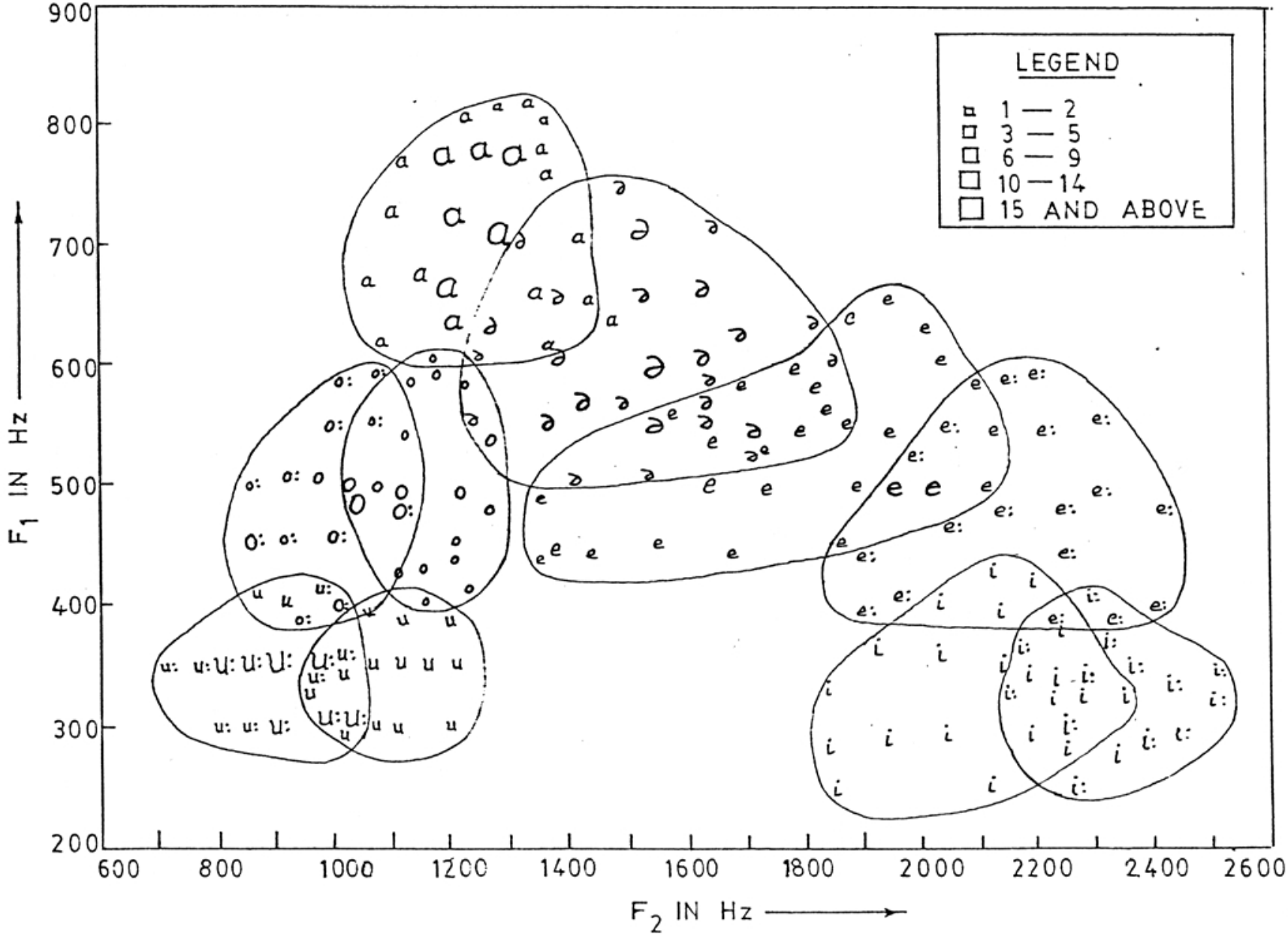

Fig.5.1 : Distribution of the Telugu vowel data (871 samples) in the F_1-F_2 plane.

Feature	Wavelength (in μm)	Colour(in the e-m spectrum)
Band 1	0.5-0.6	Green
Band 2	0.6-0.7	Red
Band 3	0.7-0.8	Near Infrared (NIR)
Band 4	0.8-1.1	Infrared

Table 5.2: Specification of the four features of the LANDSAT (MSS) data used here

The results obtained by applying the GGA and other algorithms are given in section 5.3.2.

5.2.3 The Landsat imagery data set

The data is generated with the help of the remotely-sensed images recorded by the satellite LANDSAT-V, which has two sensors:

(i) Multi-Spectral Scanner (MSS)

(ii) Thematic Mapper (TM)

The data used here is obtained by the MSS. The brightness or intensity due to any given pixel is resolved on the *electromagnetic spectrum* into four bands which are taken to be the four features. Details are given in Table 5.2. The area of the earth's surface covered by each pixel is 79m × 79m.

The data was initially bulk-corrected. The correction was incorporated in order to neutralize the various types of errors that were introduced during the acquisition of the data by the satellite platform. These errors were due to goemetric distortions (altitude, perspective, map projection, scan skew velocity, etc.), photometric distortion (motion blur, nonlinear amplitude response, defocussing, etc.) and electronic distortion(noise generated in the electronic circuitry). After the correction is done, the imagery data is properly processed for quality improvement. The processing operations include enhancement, contrast stretching, band ratioing, etc. Details of these are available in [88]. These imagery data were picked up from an area which is enclosed by the 22°0′N and 22°15′N parallels of latitude, and the 86°30′E and 86°45′E circles of longitude. A geological map of this region, provided by the Geological Survey of India, Calcutta, confirms the presence of all rock types as well as metavolcanic bodies, water bodies and vegetation. The different rock types are: Manda Granite, Quartzite, Romapahari Granite, Bhuasani Granite, Black Phyllite and Alluvium. Out of these nine possible classes, we considered initially the following seven for our problem:

- Manda Granite
- Quartzite
- Romapahari Granite
- Vegetation
- Bhuasani Granite
- Black Phyllite
- Alluvium

Initially, a total of 2600 samples was available from these classes. The breakup of this set according to the seven classes is given in Table 5.3. For ease of reference, the classes will often be referred to, in subsequent discussions, by the numbers and/or identifying letters assigned to them in the table.

A supervised classification scheme using the Bayes classifier was implemented for different training samples, but the recognition scores were found to be very poor ($\approx 40.5\%$), even with 100% training samples [88]. This poor performance is due to the enormous overlapping among all the classes. The overlapping may be because of the fact that due to several climatic and natural effects, the topmost layers of the different rocks are deformed and fresh rocky outcrops are seldom available. This was also noticed while collecting ground truth.

Since the features are highly correlated, *Principal Components Analysis* was done to reduce the four features to two principal features. This facilitates matters in a number of ways, obviously, for instance, by reducing the time required to apply the algorithm and making diagrammatic representation possible.

In order to reduce the overlapping to some extent, the sample space was made smaller by removing the classes Quartzite and Bhuasani granite. The data set too, was reduced by taking only those samples which are within a distance of 3σ (for Alluvium it is σ) from the actual means of the classes. The number of samples so extracted is 677. The distribution of these samples in the transformed two-dimensional feature space is given in figure 5.2. The GGA and other algorithms were applied to this reduced set of 677 samples.

Principal Components Analysis

This involves extracting important or strong features in an N-feature space. The original N-feature space is *transformed* so that p ($p < N$) features are strong compared to the rest. These p are treated as principal components by rejecting the remaining $(N - p)$ components. Though accuracy suffers slightly, there are several advantages as mentioned above.

Method of transformation: The entire feature space is treated as one class, the covariance matrix for which is called $\mathbf{A}_{N\times N}$. $\mathbf{B}_{N\times N}$ is the eigenvector matrix of $\mathbf{A}_{N\times N}$. If

$$\mathbf{G}_{N\times N} = \mathbf{B}^T{}_{N\times N},$$

and $\mathbf{X}_{N\times 1}$ is the original point in the N- feature space,then the corresponding point in the transformed space is

$$\mathbf{Y}_{N\times 1} = \mathbf{G}\mathbf{X}.$$

Of these, the features corresponding to the p largest eigenvalues, that is, the transformed variables Y_i having the p largest variances are called the p principal components of $\mathbf{X}$, $p < N$.

The transformation used was:

$$\begin{aligned}
Y_1 &= 0.406643X_1 - 0.180934X_2 + 0.895310X_3 + 0.018026X_4 \\
Y_2 &= 0.633041X_1 - 0.627475X_2 - 0.417871X_3 + 0.175838X_4 \\
Y_3 &= 0.550550X_1 + 0.459865X_2 - 0.143386X_3 - 0.681804X_4 \\
Y_4 &= 0.361656X_1 + 0.601716X_2 - 0.056930X_3 + 0.709859X_4
\end{aligned}$$

With this transformation, the first two components were observed to contribute to more than 90% of the total variation, and hence were taken to be the principal components.

The results obtained on applying the GGA and other algorithms to this data set are given in section 5.3.3.

Serial number given	Letter given	Name of the class	Number of samples
1.	A	Manda Granite	300
2.	B	Quartzite, Quartz-Granulate	300
3.	C	Romapahari Granite	300
4.	D	Vegetation	300
5.	E	Bhuasani granite	200
6.	X	Black Phyllite	600
7.	Y	Alluvium	600

Table 5.3: Breakup of the (extended) LANDSAT data set used here

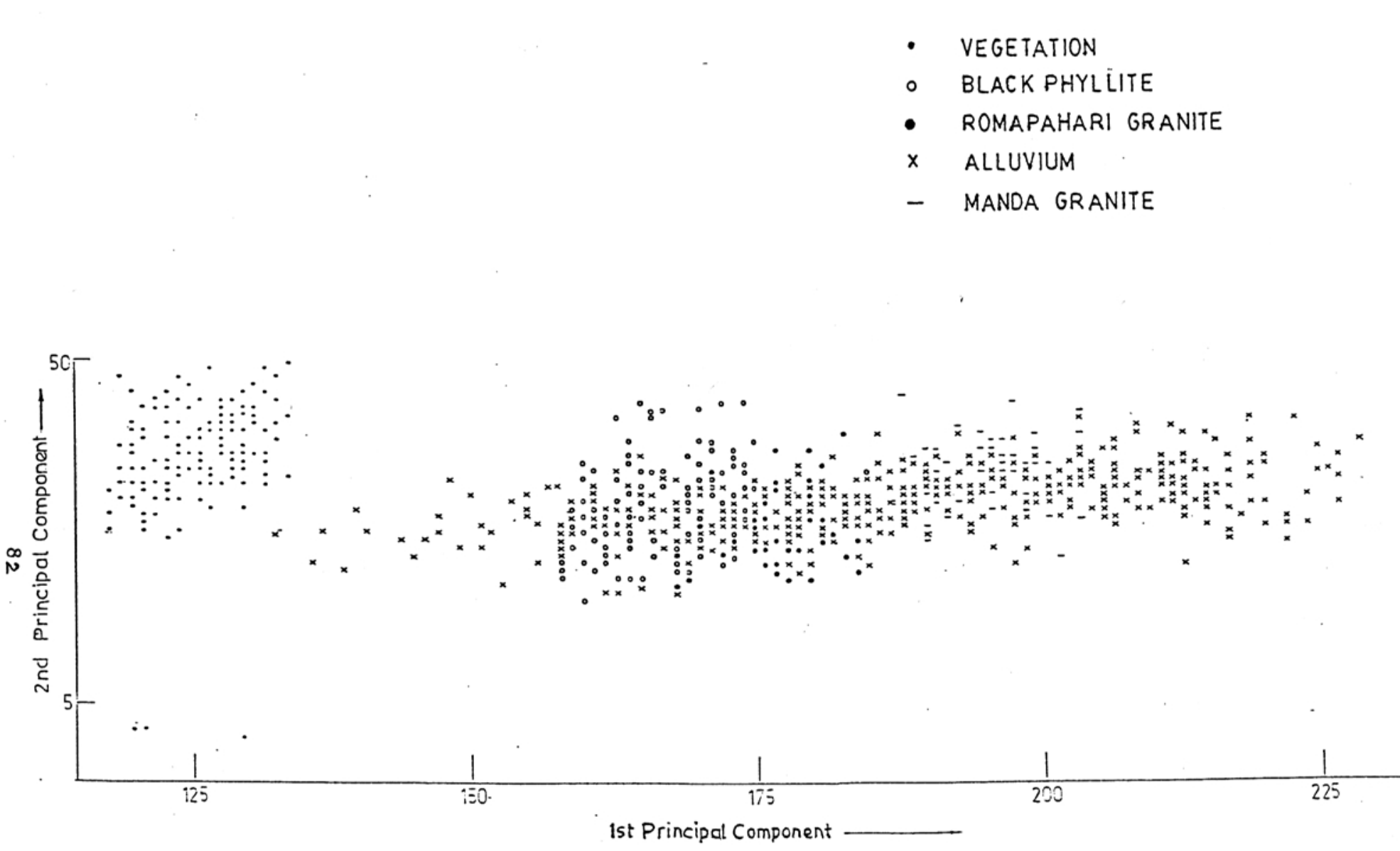

Fig.5.2 : Distribution of the LANDSAT-V data in the transformed two-dimensional feature space. (677 samples)

5.3 Experimental results

5.3.1 Results obtained in the simulated PR experiment [7,10]

The main feature of this simulated learning experiment for a three-class two-feature PR problem is that no classifier was used to obtain the labels for training samples, as in the other experiments which follow. Instead, the training set for each class was deliberately 'contaminated' with a known proportion of samples from the other two classes, the corresponding proportions α_{kj}, $k, j = 1, 2, 3$ being given in Table 5.1. As mentioned earlier on in section 5.2.1, two such data sets were prepared — ADS-I and ADS-II.

Also, as the data sets are rather small, and no classifier is being used as such, the notion of correct classification rate as a criterion for comparison does not have any meaning. For the purpose of comparison, we have used instead, the Euclidean distances of the estimates from the true values, either their individual values or their average values.

With ADS-I:

The Artificial Data Set I, whose details are given in Table 5.1, was used to verify the proposition 3.6, with the help of the estimates of $\lambda_n^{(k)}$ defined by the equation 5.1 for the best empirical choice of α. As mentioned in section 5.1, a Bayes classifier with equal *a priori* probabilities was used. Taking the first sample as the initial estimate, the GGA was applied repeatedly for different values of $\alpha \in \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9\}$. The optimum value of α was empirically found to be 0.8. The estimates obtained with this value of α are given in Tables 5.4, 5.5 and 5.6 [6,7] for the three different classes. These tables also give, for each iteration, the distances, both individual and (cumulative) average, of the estimates from the true values given in Table 5.1. These values are given for the mean vector as well as the 3-dimensional vector consisting of the distinct elements of the (raw) covariance matrix.

A careful examination of Tables 5.4, 5.5 and 5.6 reveals that the performance of the GGA is uniformly better (with respect to the 'closeness-to-the-true value' criterion) than the non-GGA for classes 1 and 2 but not for class 3. This is to be expected in view of the proposition 3.6, for it can readily be seen from Table 5.1 that while the means and covariances of classes 1 and 2 satisfy the conditions of the proposition, those of class 3 do not. That is, for $k = 1, 2$, the modified parameters $\bar{\theta}^{(k)}$ and $\bar{\Sigma}^{(k)}$ are either strictly less (for $k = 1$) or strictly greater (for $k = 2$) elementwise than the corresponding elements of the true parameters $\mu^{(k)}$ and $\Sigma^{(k)}$. However, for class 3, this is not true; while $\bar{\mu}_1^{(k)}$ is greater than $\mu_1{}^{(k)}$, $\bar{\mu}_2^{(k)}$ is less than $\mu_2{}^{(k)}$. By virtue of the proposition, therefore, the GGA-estimates for the classes 1 and 2 are expected to be *closer* to their true values in the long run, but not those for class 3.

With ADS-II:

This data set is used to compare, on the basis of the distances of the estimates from the true values, the performances of the GGA with λ-*sequence* 2 relative to the non-GGA.

The values of λ_n obtained using the estimate given by equation 5.2 with $N = 2$ and the help of tables of the incomplete beta-function [84], are given in Table 5.7 for $n = 1, 2, \ldots, 20$ [10]. The GGA and the non-GGA were implemented on the three training sets for a number of different permutations of the samples within each set. In each case, for each of the classes, we computed, after every iteration and for both algorithms, the *average* distances of the estimates from the two sets of *true* parameter values defined below in terms of their MSEs, as

$$\sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{\mu} - \mu^*)^T(\hat{\mu} - \mu^*)} = \sqrt{\text{MSE}},$$

where μ^* is the *true* value chosen. The two types of *true* parameter values considered are

(1) the sample parameter values obtained with the help of all the correctly labeled training samples of the respective classes, and

(2) the *true* population parameter values.

These *true* parameter values are given in Table 5.11 [10]. The *average* distance ($\sqrt{\text{MSE}}$) defined above is simply the square root of the arithmetic mean of the squared Euclidean distances of the estimates from their *true* values. These individual Euclidean distances too, were taken into account for purposes of comparison of the two algorithms.

It was found, in a large majority of cases, that the distances, particularly the *average* distances, for the GGA-estimates were smaller than those for the non-GGA-estimates. This was found to be strictly true in the cases where the sample *true* values were the values used as the standard. As a typical example, the complete results for one particular ordering of the training sets using the sample *true* values as the standard, are given in Tables 5.8, 5.9 and 5.10 [10] respectively. The initial and final estimates, as obtained by both the algorithms, are given in Table 5.11 [10]. Also, as mentioned earlier in section 5.1, we have estimated the *uncorrected (raw)* second-order moments rather than the central second-order moments. In other words, we have estimated $\mathcal{E}(\mathbf{X}\mathbf{X}^T)$ rather than $\mathcal{E}(\mathbf{X} - \mu^*)(\mathbf{X} - \mu^*)^T$. This had to be done as the condition (CP1) of the proposition 4.1 would not have been satisfied otherwise.

5.3.2 Results obtained with the Telugu vowel data set [7,8]

Here too, the experiment was done in two parts —

1. the first, for examining the effect of λ-sequence 1 alone,
2. the second, for examinig the effect of λ-sequence 2, and comparing the performances of both λ-sequences.

In both parts, comparisons were made with the non-GGA and/or the fully-supervised scheme, and the GGA with constant λ-values.

Labeling was done with a Bayes classifier based on the assumption of independence of features, that is, the covariance matrix was assumed to be diagonal. The block diagram of the dynamic self-supervised recognition system based on the GGA is given in figure 5.3.

Part 1

Here, we are interested mainly in studying :

(a) the adaptive efficiency of the system in recognising vowel sounds starting with *poor* (non-appropriate) estimates of the parameters representing the classes;

(b) the effect of α (i.e., the weighting co-efficient for determining the dimension of guard zone) in equation 5.1 on the performance of the system with an attempt to determine experimentally its optimum value; and

(c) the effect of 'dynamic behavior of the guard zone' in acting as a supervisor on the decision of the classifier.

The first of these points involved the learning of μ and Σ with only five samples selected randomly from the utterances of

(i) a single speaker,

and

(ii) three speakers

so that the initial estimates may be designated as *very weak* and *not too weak*, say, respectively. Recognition efficiency obtained with such weak representative parameters was compared with that of a fully supervised system for different (random) orderings of the input sequence.

The above experiment was then repeated for different values of α, namely, 0.1, 0.2, 0.3, ..., 0.7, 0.8, 0.9 for demonstrating the second point of interest.

In order to exhibit the third point of interest, the performance of the classifier for the aforesaid cases was compared with those obtained when the parameter λ is taken to be fixed throughout the learning process. Two such fixed values considered here are

(i) λ_1 (i.e., the value of generated by the system at $n = 1$), and

(ii) 1/2 (an optimum value obtained by Pal et al. [53] and Pal [89] with fixed guard zone dimension for two different types of classifiers).

In other words, this part also gives a comparison of the proposed algorithm (GGA) with the existing ones based on similar concepts.

Table 5.4 : Learning of Means and Covariances of Class 1 using GGA and non-GGA

ser. num.	sample	true class	d(.,.)	lamda	updt?	GGA-estimates of mean vector		GGA-estimates of covariance matrix(raw)		distance from true parameter values of the GGA-estimates of means indiv.	of means average	of dispersions indiv.	of dispersions average
1	11.20 15.09	1	0.00	0.00	-	11.200	15.088	125.4469	168.9856				
2	12.38 13.80	1	0.00	0.00	-	11.788	14.446	168.9856 139.3107	227.6351 169.9111	1.204	1.204	28.656	28.656
3	10.82 14.34	1	3.35	3.65	Y	11.467	14.412	169.9111 131.9301	209.0852 165.0289	1.872	1.574	47.023	38.938
4	10.58 16.67	1	4.74	6.39	Y	11.245	14.977	165.0289 126.9330	207.9721 167.8718	1.581	1.576	40.412	39.435
5	8.91 15.46	1	4.56	4.99	Y	10.777	15.073	167.8718 117.4074	225.4735 161.8245	1.246	1.500	29.688	37.238
6	10.38 13.25	1	2.50	2.44	N	10.777	15.073	161.8245 117.4074	228.1524 161.8245	0.781	1.387	18.100	34.277
7	5.41 6.48	2	14.58	13.25	N	10.777	15.073	161.8245 117.4074	228.1524 161.8245	0.781	1.305	18.100	32.151
8	9.42 11.41	1	5.45	5.11	N	10.777	15.073	161.8245 117.4074	228.1524 161.8245	0.781	1.244	18.100	30.542
9	8.48 14.39	1	3.02	3.13	Y	10.522	14.997	161.8245 112.3541	228.1524 157.4067	0.781	1.196	18.100	29.277
10	8.07 10.19	3	0.74	3.71	Y	10.277	14.516	157.4067 107.6277	225.8188 149.8856	0.522	1.141	12.973	27.939
11	9.66 13.66	1	0.16	0.76	Y	10.221	14.438	149.8856 106.3306	213.6163 148.2601	0.557	1.097	20.040	27.253
12	10.33 14.48	1	0.07	0.09	Y	10.230	14.442	148.2601 106.3605	211.1649 148.3653	0.603	1.061	22.402	26.848
13	6.06 9.49	3	1.33	5.12	Y	9.909	14.061	148.3653 101.0079	211.0301 141.3792	0.604	1.031	22.521	26.514
14	9.41 14.72	1	0.74	0.66	N	9.909	14.061	141.3792 101.0079	201.7237 141.3792	0.944	1.024	33.090	27.077
15	10.91 12.82	1	1.44	1.26	N	9.909	14.061	141.3792 101.0079	201.7237 141.3792	0.944	1.019	33.090	27.550
16	12.20 14.37	1	1.72	1.83	Y	10.052	14.080	141.3792 103.9917	201.7237 143.4929	0.944	1.014	33.090	27.954
17	11.99 16.39	1	0.65	2.38	Y	10.166	14.215	143.4929 106.3263	202.0130 146.6062	0.922	1.009	32.149	28.234
18	13.60 16.48	1	1.83	3.34	Y	10.357	14.341	146.6062 110.6910	205.9249 150.9139	0.802	0.998	27.807	28.209
19	11.03 12.96	1	1.23	1.20	N	10.357	14.341	150.9139 110.6910	209.5808 150.9139	0.749	0.985	24.674	28.024
20	11.41 16.42	1	0.42	1.83	Y	10.409	14.445	150.9139 111.6657 152.7373	209.5808 152.7373 212.5871	0.749	0.974	24.674	27.858
										0.689	0.962	22.188	27.602

Table 5.4 contd.

non-GGA estimates of mean vector		non-GGA estimates of covariance matrix(raw)		distances from true parameter values of the non-GGA estimates of means		of dispersions	
				indiv.	average	indiv.	average
11.200	15.088	125.4469	168.9856				
		168.9856	227.6351	1.204	1.204	28.656	28.656
11.788	14.446	139.3107	169.9111				
		169.9111	209.0852	1.872	1.574	47.023	38.938
11.467	14.412	131.9301	165.0289				
		165.0289	207.9721	1.581	1.576	40.412	39.435
11.245	14.977	126.9330	167.8718				
		167.8718	225.4735	1.246	1.500	29.688	37.238
10.777	15.073	117.4074	161.8245				
		161.8245	228.1524	0.781	1.387	18.100	34.277
10.712	14.768	115.8125	157.7820				
		157.7820	219.3767	0.749	1.302	19.575	32.294
9.954	13.584	103.4459	140.2449				
		140.2449	194.0284	1.417	1.319	40.708	33.626
9.887	13.312	101.6067	136.1449				
		136.1449	186.0379	1.692	1.371	49.586	36.010
9.731	13.432	98.3090	134.5804				
		134.5804	188.3837	1.591	1.397	48.125	37.549
9.565	13.107	94.9871	129.3419				
		129.3419	179.9247	1.942	1.461	58.263	40.105
9.574	13.158	94.8391	129.5841				
		129.5841	180.5361	1.891	1.505	57.633	42.002
9.637	13.268	95.8267	131.2455				
		131.2455	182.9538	1.770	1.529	54.652	43.198
9.362	12.977	91.2844	125.5764				
		125.5764	175.8071	2.121	1.583	64.082	45.148
9.365	13.101	91.0835	126.4963				
		126.4963	178.7267	2.002	1.616	61.139	46.474
9.468	13.083	92.9530	127.3942				
		127.3942	177.7748	1.989	1.644	61.288	47.605
9.639	13.163	96.4402	130.3820				
		130.3820	179.5610	1.872	1.659	58.018	48.321
9.777	13.353	99.2190	134.2665				
		134.2665	184.7935	1.662	1.659	51.504	48.514
9.989	13.527	103.9786	139.2598				
		139.2598	189.6234	1.473	1.649	45.219	48.337
10.044	13.497	104.9119	139.4564				
		139.4564	188.4855	1.504	1.642	46.288	48.232
10.112	13.643	106.1756	141.8527				
		141.8527	192.5466	1.361	1.629	41.827	47.932

Table 5.5 : Learning of Means and Covariances of Class 2 using GGA and non-GGA

ser. num.	sample		true class	d(.,.)	lamda	updt?	GGA-estimates of mean vector		GGA-estimates of covariance matrix(raw)		distance from true parameter values of the GGA-estimates of means		of dispersions	
											indiv.	average	indiv.	average
1	4.68	4.42	2	0.42	1.83	-	4.682	4.416	21.9211	20.6757				
2	5.56	4.54	2	0.42	1.83	-	5.120	4.478	20.6757 26.4062	19.5011 22.9545	0.665	0.665	12.611	12.611
3	6.19	4.72	2	2.46	3.32	Y	5.476	4.560	22.9545 30.3679	20.0563 25.0470	0.536	0.604	9.534	11.179
4	5.37	4.02	2	1.64	1.59	N	5.476	4.560	25.0470 30.3679	20.8096 25.0470	0.648	0.619	8.304	10.310
5	0.51	4.19	2	8.28	14.46	Y	4.483	4.487	25.0470 24.3464	20.8096 20.4654	0.648	0.626	8.304	9.847
6	2.57	4.29	2	0.93	1.93	Y	4.165	4.454	20.4654 21.3929	20.1656 18.8949	0.729	0.648	10.967	10.081
7	4.11	3.68	2	0.92	0.84	N	4.165	4.454	18.8949 21.3929	19.8720 18.8949	0.998	0.718	13.359	10.697
8	2.27	6.19	2	2.70	2.79	Y	3.928	4.671	18.8949 19.3618	19.8720 18.2874	0.998	0.765	13.359	11.117
9	6.88	3.11	2	2.12	2.38	Y	4.256	4.497	18.2874 22.4759	22.1744 18.6327	1.122	0.818	13.585	11.454
10	3.04	9.57	3	3.80	3.55	N	4.256	4.497	18.6327 22.4759	20.7839 18.6327	0.898	0.827	12.272	11.548
11	4.37	4.07	2	0.32	0.30	N	4.256	4.497	18.6327 22.4759	20.7839 18.6327	0.898	0.834	12.272	11.622
12	9.78	8.27	1	3.67	4.56	Y	4.717	4.812	18.6327 28.5765	20.7839 23.8237	0.898	0.840	12.272	11.683
13	7.96	5.73	2	5.71	19.77	Y	4.966	4.882	23.8237 31.2474	24.7555 25.4967	0.340	0.811	4.425	11.258
14	2.98	5.00	2	2.36	14.93	Y	4.824	4.891	25.4967 29.6489	25.3751 24.7399	0.123	0.780	4.294	10.882
15	4.50	8.89	3	14.40	11.82	N	4.824	4.891	24.7399 29.6489	25.3511 24.7399	0.207	0.753	3.715	10.533
16	4.26	5.44	2	2.10	2.31	Y	4.789	4.925	24.7399 28.9322	25.3511 24.6434	0.207	0.730	3.715	10.221
17	4.25	9.97	3	16.58	13.91	N	4.789	4.925	24.6434 28.9322	25.6163 24.6434	0.224	0.709	3.403	9.933
18	4.99	3.09	2	6.01	5.05	N	4.789	4.925	24.6434 28.9322	25.6163 24.6434	0.224	0.690	3.403	9.672
19	4.32	1.54	2	11.38	9.37	N	4.789	4.925	24.6434 28.9322	25.6163 24.6434	0.224	0.672	3.403	9.433
20	5.23	1.07	2	12.64	10.62	N	4.789	4.925	24.6434 28.9322 24.6434	25.6163 24.6434 25.6163	0.224	0.656	3.403	9.215
											0.224	0.642	3.403	9.014

Table 5.5 contd.

non-GGA estimates of mean vector		non-GGA estimates of covariance matrix(raw)		distances from true parameter values of the non-GGA estimates of means		of dispersions	
				indiv.	average	indiv.	average
4.682	4.416	21.9211	20.6757	0.665	0.665	12.611	12.611
		20.6757	19.5011				
5.120	4.478	26.4062	22.9545	0.536	0.604	9.534	11.179
		22.9545	20.0563				
5.476	4.560	30.3679	25.0470	0.648	0.619	8.304	10.310
		25.0470	20.8096				
5.450	4.426	29.9852	24.1848	0.729	0.648	9.436	10.099
		24.1848	19.6513				
4.462	4.379	24.0402	19.7756	0.822	0.687	12.131	10.537
		19.7756	19.2390				
4.147	4.364	21.1377	18.3201	1.064	0.762	14.299	11.251
		18.3201	19.0998				
4.141	4.267	20.5265	17.8627	1.129	0.825	15.397	11.932
		17.8627	18.3080				
3.907	4.507	18.6037	17.3842	1.199	0.880	15.272	12.399
		17.3842	20.8059				
4.238	4.352	21.8021	17.8299	1.001	0.895	13.863	12.570
		17.8299	19.5675				
4.118	4.873	20.5454	18.9538	0.891	0.894	10.633	12.390
		18.9538	26.7608				
4.141	4.800	20.4121	18.8453	0.882	0.893	11.031	12.272
		18.8453	25.8309				
4.611	5.089	26.6847	24.0186	0.399	0.863	2.544	11.773
		24.0186	29.3819				
4.868	5.138	29.5011	25.6766	0.191	0.831	1.061	11.315
		25.6766	29.6456				
4.733	5.129	28.0273	24.9069	0.296	0.804	1.027	10.907
		24.9069	29.3166				
4.717	5.379	27.5083	25.9116	0.473	0.787	4.025	10.588
		25.9116	32.6255				
4.689	5.383	26.9254	25.7419	0.493	0.772	4.082	10.303
		25.7419	32.4360				
4.663	5.653	26.4045	26.7217	0.735	0.769	8.010	10.182
		26.7217	36.3799				
4.681	5.511	26.3187	26.0942	0.602	0.761	6.564	10.015
		26.0942	34.8906				
4.662	5.302	25.9167	25.0707	0.453	0.748	5.194	9.821
		25.0707	33.1788				
4.691	5.090	25.9906	24.0982	0.322	0.733	4.064	9.615
		24.0982	31.5775				

Table 5.6 : Learning of Means and Covariances of Class 3 using GGA and non-GGA

ser. num.	sample		true class	d(.,.)	lamda	updt?	GGA-estimates of mean vector		GGA-estimates of covariance matrix(raw)		distance from true parameter values of the GGA-estimates of means		of dispersions	
											indiv.	average	indiv.	average
1	4.30	8.51	3	12.64	10.62	-	4.305	8.511	18.5294	36.6361				
2	4.72	8.25	3	12.64	10.62	-	4.511	8.380	36.6361 20.3957	72.4361 37.7789	1.643	1.643	34.908	34.908
3	3.59	11.43	3	9.00	15.65	Y	4.203	9.398	37.7789 17.8855	70.2424 38.8563	1.692	1.668	35.933	35.425
4	4.56	11.88	3	8.46	15.74	Y	4.293	10.017	38.8563 18.6216	90.4062 42.6926	0.999	1.479	20.205	31.188
5	7.01	7.01	3	6.34	8.88	Y	4.836	9.416	42.6926 24.7196	103.0644 43.9790	0.707	1.329	13.010	27.782
6	2.56	13.24	3	2.15	5.27	Y	4.457	10.053	43.9790 21.6905	92.2789 42.2933	0.607	1.219	12.963	25.516
7	7.61	12.16	1	5.59	4.79	N	4.457	10.053	42.2933 21.6905	106.1049 42.2933	0.546	1.135	11.121	23.731
8	5.03	7.03	2	1.82	3.88	Y	4.528	9.676	42.2933 22.1367	106.1049 41.4257	0.546	1.071	11.121	22.369
9	8.67	10.02	1	41.91	44.10	Y	4.989	9.713	41.4257 28.0388	99.0264 46.4776	0.573	1.022	11.619	21.324
10	2.48	10.92	3	6.64	16.02	Y	4.738	9.834	46.4776 25.8499	99.1712 44.5373	0.287	0.968	4.987	20.173
11	2.54	9.12	3	13.59	12.36	N	4.738	9.834	44.5373 25.8499	101.1728 44.5373	0.310	0.924	6.362	19.243
12	5.58	11.55	3	11.98	10.22	N	4.738	9.834	44.5373 25.8499	101.1728 44.5373	0.310	0.886	6.362	18.448
13	5.50	10.34	3	5.92	4.88	N	4.738	9.834	44.5373 25.8499	101.1728 44.5373	0.310	0.853	6.362	17.758
14	9.53	9.07	3	18.84	26.01	Y	5.080	9.779	44.5373 30.4972	101.1728 47.5341	0.310	0.824	6.362	17.152
15	6.71	10.64	3	8.80	7.49	N	5.080	9.779	47.5341 30.4972	99.8240 47.5341	0.235	0.796	3.533	16.555
16	4.27	7.62	2	10.55	9.40	N	5.080	9.779	47.5341 30.4972	99.8240 47.5341	0.235	0.772	3.533	16.020
17	4.86	11.12	3	4.03	5.54	Y	5.068	9.858	47.5341 30.0947	99.8240 47.9198	0.235	0.749	3.533	15.536
18	6.57	11.69	3	10.90	8.92	N	5.068	9.858	47.9198 30.0947	101.2279 47.9198	0.157	0.728	2.078	15.081
19	4.95	6.46	2	11.47	12.80	Y	5.062	9.679	47.9198 29.8014	101.2279 47.0809	0.157	0.709	2.078	14.664
20	6.62	10.81	3	10.78	8.85	N	5.062	9.679	47.0809 29.8014	98.0951 47.0809	0.327	0.694	5.271	14.324
									47.0809	98.0951	0.327	0.680	5.271	14.011

Table 5.6 contd.

non-GGA estimates of mean vector		non-GGA estimates of covariance matrix(raw)		distances from true parameter values of the non-GGA estimates of means		of dispersions	
				indiv.	average	indiv.	average
4.305	8.511	18.5294	36.6361				
		36.6361	72.4361	1.643	1.643	34.908	34.908
4.511	8.380	20.3957	37.7789				
		37.7789	70.2424	1.692	1.668	35.933	35.425
4.203	9.398	17.8855	38.8563				
		38.8563	90.4062	0.999	1.479	20.205	31.188
4.293	10.017	18.6216	42.6926				
		42.6926	103.0644	0.707	1.329	13.010	27.782
4.836	9.416	24.7196	43.9790				
		43.9790	92.2789	0.607	1.219	12.963	25.516
4.457	10.053	21.6905	42.2933				
		42.2933	106.1049	0.546	1.135	11.121	23.731
4.907	10.353	26.8682	49.4686				
		49.4686	112.0547	0.365	1.060	9.592	22.268
4.922	9.938	26.6673	47.7042				
		47.7042	104.2325	0.099	0.992	3.782	20.873
5.339	9.947	32.0660	52.0584				
		52.0584	103.7988	0.343	0.942	3.776	19.719
5.053	10.044	29.4744	49.5600				
		49.5600	105.3377	0.069	0.894	2.461	18.723
4.825	9.960	27.3826	47.1632				
		47.1632	103.3274	0.179	0.854	3.214	17.878
4.888	10.093	27.6946	48.6017				
		48.6017	105.8287	0.145	0.819	3.671	17.150
4.935	10.112	27.8881	49.2345				
		49.2345	105.9110	0.129	0.788	3.604	16.507
5.263	10.037	32.3898	51.8958				
		51.8958	104.2236	0.266	0.762	3.949	15.942
5.359	10.077	33.2286	53.1926				
		53.1926	104.8218	0.368	0.742	5.597	15.469
5.292	9.924	32.2935	51.9025				
		51.9025	101.8956	0.301	0.723	3.861	15.009
5.266	9.994	31.7853	52.0313				
		52.0313	103.1777	0.266	0.704	3.522	14.586
5.339	10.088	32.4173	53.4088				
		53.4088	105.0428	0.350	0.689	5.427	14.232
5.318	9.897	32.0018	52.2809				
		52.2809	101.7093	0.335	0.675	4.054	13.884
5.383	9.943	32.5907	53.2436				
		53.2436	102.4679	0.388	0.664	5.000	13.579

Iteration number n	λ-value λ_n
1	-
2	-
3	11.2450
4	5.5656
5	4.4333
6	3.8401
7	3.5659
8	3.3988
9	3.2876
10	3.2104
11	3.1556
12	3.1153
13	3.0839
14	3.0596
15	3.0415
16	3.0284
17	3.0165
18	3.0078
19	3.0022
20	2.9968

Table 5.7: Table of λ-values for $n = 1, 2, \ldots, 20$ when the feature vector dimension (N) is 2

Table 5.8 : Learning of Means and Covariances for Class 1 Using GGA and non-GGA

sample no.	training sample		true class	distance	updated?	Euclidean distances from 'true' sample values of: GGA-estimates of mean vector indiv.	GGA-estimates of mean vector ave.	GGA-estimates of covariance indiv.	GGA-estimates of covariance ave.	non-GGA estimates of mean vector indiv.	non-GGA estimates of mean vector ave.	non-GGA estimates of covariance indiv.	non-GGA estimates of covariance ave.
1	9.41	14.72	1			1.358	1.358	34.219	34.219	1.358	1.358	34.219	34.219
2	11.20	15.09	1			0.592	1.047	14.265	26.215	0.592	1.047	14.265	26.215
3	12.38	13.80	1	4.12	Y	0.248	0.867	5.975	21.681	0.248	0.867	5.975	21.681
4	10.82	14.34	1	0.57	Y	0.206	0.758	4.939	18.938	0.206	0.758	4.939	18.938
5	10.58	16.67	1	5.40	N	0.206	0.684	4.939	17.082	0.433	0.705	12.468	17.832
6	8.91	15.46	1	8.47	N	0.206	0.630	4.939	15.723	0.539	0.680	14.408	17.309
7	10.38	13.25	1	6.61	N	0.206	0.588	4.939	14.676	0.334	0.642	8.406	16.337
8	5.41	6.48	2	42.40	N	0.206	0.555	4.939	13.839	1.167	0.729	27.726	18.155
9	9.42	11.41	1	19.76	N	0.206	0.528	4.939	13.151	1.388	0.828	35.183	20.749
10	8.48	14.39	1	8.46	N	0.206	0.505	4.939	12.573	1.416	0.904	36.373	22.799
11	8.07	10.19	3	30.29	N	0.206	0.486	4.939	12.080	1.737	1.009	45.636	25.727
12	9.66	13.66	1	7.60	N	0.206	0.469	4.939	11.654	1.706	1.084	45.409	27.902
13	10.33	14.48	1	2.33	Y	0.084	0.451	2.807	11.224	1.599	1.132	42.795	29.318
14	6.06	9.49	3	73.16	N	0.084	0.435	2.807	10.841	1.975	1.212	52.343	31.525
15	10.91	12.82	1	16.95	N	0.084	0.421	2.807	10.499	1.918	1.271	51.555	33.238
16	12.20	14.37	1	4.69	N	0.084	0.408	2.807	10.189	1.746	1.306	47.052	34.265
17	11.99	16.39	1	10.24	N	0.084	0.396	2.807	9.909	1.512	1.319	39.983	34.628
18	13.60	16.48	1	1.65	Y	0.622	0.412	18.223	10.544	1.244	1.315	31.720	34.472
19	11.03	12.96	1	0.72	Y	0.507	0.418	13.452	10.717	1.235	1.311	32.083	34.351
20	11.41	16.42	1	0.74	Y	0.592	0.428	17.098	11.123	1.077	1.300	27.268	34.032

Table 5.9 : Learning of Means and Covariances for Class 1 Using GGA and non-GGA

sample no.	training sample	true class	distance	updated?	Euclidean distances from 'true' sample values of: GGA-estimates of: mean vector: indiv.	GGA-estimates of: mean vector: ave.	GGA-estimates of: covariance: indiv.	GGA-estimates of: covariance: ave.	non-GGA estimates of: mean vector: indiv.	non-GGA estimates of: mean vector: ave.	non-GGA estimates of: covariance: indiv.	non-GGA estimates of: covariance: ave.
1	5.56 4.54	2			1.144	1.144	10.499	10.499	1.144	1.144	10.499	10.499
2	4.68 4.42	2			0.730	0.959	5.900	8.516	0.730	0.959	5.900	8.516
3	6.19 4.72	2	1.73	Y	1.079	1.001	10.056	9.058	1.079	1.001	10.056	9.058
4	5.37 4.02	2	0.96	Y	1.000	1.000	8.966	9.035	1.000	1.000	8.966	9.035
5	0.51 4.19	2	9.34	N	1.000	1.000	8.966	9.021	0.315	0.906	1.975	8.129
6	2.57 4.29	2	11.70	N	1.000	1.000	8.966	9.012	0.472	0.849	2.531	7.493
7	4.11 3.68	2	8.88	N	1.000	1.000	8.966	9.005	0.423	0.802	3.019	7.030
8	2.27 6.19	2	3.18	Y	0.769	0.974	6.648	8.745	0.750	0.796	5.568	6.864
9	6.88 3.11	2	1.36	Y	0.774	0.954	7.088	8.577	0.396	0.762	2.151	6.511
10	3.04 9.57	3	12.41	N	0.774	0.938	7.088	8.440	0.897	0.776	8.992	6.800
11	4.37 4.07	2	0.77	Y	0.645	0.915	5.492	8.216	0.821	0.781	8.166	6.936
12	9.78 8.27	1	5.64	N	0.645	0.895	5.492	8.024	1.024	0.804	12.932	7.618
13	7.96 5.73	2	3.77	N	0.645	0.879	5.492	7.858	1.126	0.833	14.872	8.401
14	2.98 5.00	2	8.88	N	0.645	0.864	5.492	7.714	1.081	0.853	13.688	8.884
15	4.50 8.89	3	17.30	N	0.645	0.851	5.492	7.586	1.325	0.892	16.774	9.613
16	4.26 5.44	2	5.40	N	0.645	0.840	5.492	7.472	1.325	0.925	16.403	10.171
17	4.25 9.97	3	9.93	N	0.645	0.830	5.492	7.370	1.591	0.977	20.214	11.018
18	4.99 3.09	2	1.88	Y	0.560	0.817	4.420	7.238	1.451	1.009	18.604	11.571
19	4.32 5.54	2	2.08	Y	0.559	0.806	4.413	7.117	1.450	1.037	18.301	12.020
20	5.23 4.07	2	0.66	Y	0.562	0.795	4.349	7.005	1.382	1.057	17.406	12.345

Table 5.10: Learning of Means and Covariances for Class 1 Using GGA and non-GGA

sample no.	training sample	true class	distance	update?	Euclidean distances from 'true' sample values of: GGA-estimates of: mean vector: indiv.	GGA-estimates of: mean vector: ave.	GGA-estimates of: covariance: indiv.	GGA-estimates of: covariance: ave.	non-GGA estimates of: mean vector: indiv.	non-GGA estimates of: mean vector: ave.	non-GGA estimates of: covariance: indiv.	non-GGA estimates of: covariance: ave.
1	4.30 8.51	3			2.041	2.041	42.445	42.445	2.041	2.041	42.445	42.445
2	6.62 10.81	3			0.779	1.545	15.615	31.980	0.779	1.545	15.615	31.980
3	5.50 10.34	3	2.03	Y	0.587	1.306	11.727	26.975	0.587	1.306	11.727	26.975
4	6.57 11.69	3	2.22	Y	0.606	1.171	9.167	23.806	0.606	1.171	9.167	23.806
5	4.95 6.46	2	7.97	N	0.606	1.082	9.167	21.684	0.924	1.126	15.521	22.396
6	3.59 11.43	3	3.28	Y	0.254	0.993	4.947	19.898	0.510	1.049	9.316	20.795
7	4.56 11.88	3	1.74	Y	0.408	0.932	8.442	18.696	0.212	0.974	4.227	19.319
8	4.72 8.25	3	8.21	N	0.408	0.884	8.442	17.741	0.452	0.925	8.964	18.347
9	4.27 4.62	2	36.76	N	0.408	0.844	8.442	16.962	1.049	0.940	18.380	18.351
10	4.86 11.12	3	0.14	Y	0.455	0.814	9.258	16.355	0.874	0.933	15.390	18.076
11	5.03 4.03	2	20.31	N	0.455	0.788	9.258	15.842	1.367	0.981	22.870	18.563
12	8.61 14.16	1	23.09	N	0.455	0.766	9.258	15.401	0.944	0.978	12.136	18.115
13	6.71 10.64	3	1.42	Y	0.474	0.747	9.320	15.021	0.879	0.970	11.054	17.672
14	8.67 10.02	1	3.12	N	0.474	0.731	9.320	14.688	0.946	0.969	12.134	17.336
15	7.01 7.01	3	10.47	N	0.474	0.717	9.320	14.392	1.139	0.981	15.405	17.214
16	2.54 9.12	3	2.16	Y	0.270	0.697	4.973	13.990	1.066	0.986	15.404	17.106
17	5.58 11.55	3	0.37	Y	0.343	0.682	6.735	13.671	0.949	0.984	13.258	16.904
18	2.56 13.24	3	3.05	N	0.343	0.667	6.735	13.380	0.695	0.970	8.623	16.553
19	2.48 10.92	3	1.58	Y	0.463	0.658	7.445	13.135	0.598	0.955	7.569	16.205
20	9.53 9.07	3	2.83	Y	0.238	0.644	4.435	12.840	0.693	0.943	9.223	15.929

Table 5.11: True and Estimated Parameter Values for the Three Classes

	CLASS 1		CLASS 2		CLASS 3	
Population values of						
Mean vector	10.000	15.000	5.000	5.000	5.000	10.000
Covariance matrix* (uncorrected)	103.000	152.000	29.000	25.000	30.000	49.000
	152.000	233.000	25.000	29.000	49.000	103.000
True sample estimates of						
Mean vector	10.747	14.512	4.516	4.069	5.142	10.372
Covariance matrix* (uncorrected)	117.200	156.605	23.536	18.100	30.038	52.372
	156.605	212.656	18.100	18.323	52.372	110.139
Initial estimates of						
Mean vector	9.406	14.720	5.558	4.540	4.305	8.511
Covariance matrix* (uncorrected)	88.471	138.456	30.891	25.233	18.529	36.636
	138.456	216.682	25.233	20.612	36.636	72.436
Final GGA-estimates of						
Mean vector	11.272	14.787	4.986	4.377	5.237	10.590
Covariance matrix* (uncorrected)	128.468	167.198	26.249	21.061	31.055	55.305
	167.198	219.949	21.061	19.993	55.305	113.306
Final non-GGA estimates of						
Mean vector	10.112	13.643	4.691	5.440	5.433	9.743
Covariance matrix* (uncorrected)	106.176	141.853	25.991	25.748	33.402	53.318
	141.853	192.547	25.748	33.765	53.318	101.604

*The matrix E (XX')

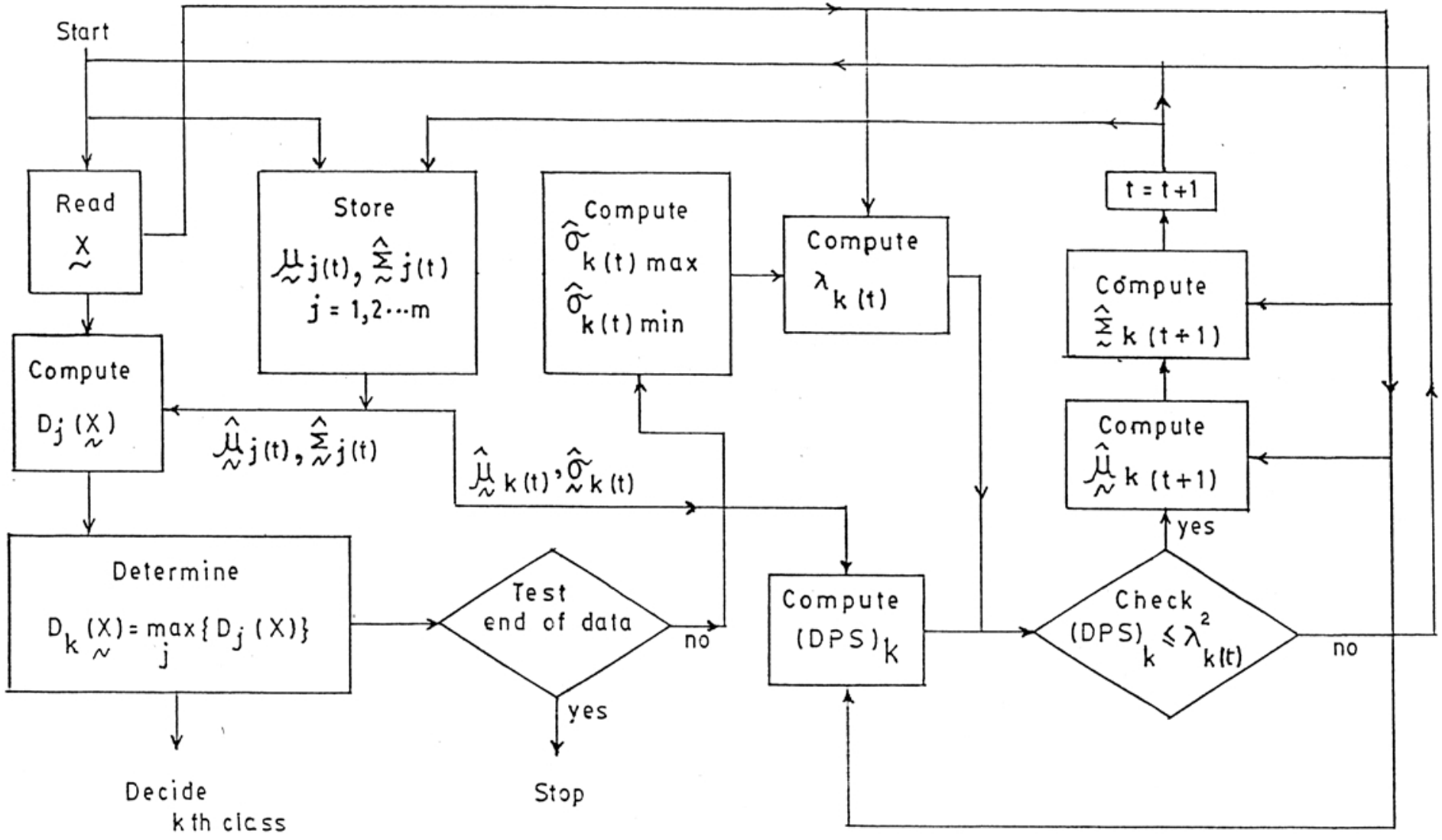

Fig.5.3 : Block diagram of the dynamic selfsupervised recognition system based on a Bayes classifier.

Since the performance of an adaptive system depends much on the sequence of incoming samples, the experiment was repeated several times for different orders of appearance of the events in the sample space. Figures 5.4 and 5.5 [7] illustrate, for three such typical instances, the variation of cumulative recognition score after every 100 samples for different values of α. Figure 5.4 corresponds to cases when the training set of 5 samples is taken only from a single speaker whereas, the results corresponding to cases when all the speakers are considered for drawing those 5 samples are depicted in Figure 5.5. Results obtained with self-supervised learning as performed by the GGA are compared in each case with those for fully-supervised (FS) case.

As expected, Figure 5.5 (with *not too weak* initial parameters) shows higher recognition score than Figure 5.4 where initial class parameters were selected to be *very weak*. With such *very weak* representative parameters, the system could not improve significantly its performance even for the fully-supervised case (Figure 5.4). This is not the case with Figure 5.5, where fully- supervised learning is found to provide an overall increase (8%) in recognition score.

From the two figures it is seen that when the initial estimates are *very weak*, good system performance is observed for values of α ranging from 0.7 to 0.9 i.e., high λ-value whereas, the range is found to be 0.1 to 0.3 i.e., low λ-value for *not too weak* initial estimates (Figure 5.5). This means, when the initial estimates are not so bad, a very lenient supervisor on lifting a strict check on the incoming samples may affect the system performance by shifting the mean and co-variance values away from their true ones. On the other hand, the guard zone needs to be flexed more, for the bad estimates, in order to strengthen the estimates by allowing higher proportion of correct to incorrect samples more available.

It is also to be noted from Figure 5.4(b) that the performance corresponding to higher α-value (i.e., higher λ-value) is better even than the case of FS learning, while the results corresponding to low α-value are the worst among the three instances (Figures 5.4(a)-(c)). Further investigations revealed that the first few sets of input sequence provided here very good proportion of correct to incorrect samples. As a result, incorporating/discarding them by expanding/shrinking the guard zone improved/declined the estimates, and hence the recognition score, significantly.

Finally, the effect of dynamic property of the supervisory program is demonstrated through Table 5.12 [8]. Here we have considered, as a typical illustration, only three samples from class *o*: which were correctly identified by the classifier. It is seen that the higher order knowledge (obtained from the input sequence) of the supervisor enables the sample which has the largest distance (Euclidean) to get selected for the updating procedure while rejecting the remaining two even though they have smaller distances. Had the value of λ_n been fixed at 1/2 and 2 throughout the learning proces, the response would have been 'reject' and 'accept' respectively in all the three instances.

The superiority of the dynamic λ-value over the fixed λ-value in improving the GGA's performance is illustrated in Figure 5.6 [8] when λ_n is kept fixed at λ_1 (the initial value generated by the system) and, $1/\sqrt{2}$ and 1/2. Here the input sequences for Figures 5.6(a) and 5.6(b) are the same as in Figures 5.4(a) and 5.4(b). The results corresponding to *very weak* initial estimates (estimated with five samples taken from a single speaker) only are shown here as an illustration.

True class	Recognised class C	Distance from C	λ	Response	λ	Response	λ	Response
o:	o:	9.35×10	0.67	reject	0.5	reject	2.0	accept
o:	o:	2.16×10^2	1.58	accept	0.5	reject	2.0	accept
o:	o:	6.9×10	0.49	reject	0.5	reject	2.0	accept

Table 5.12: A sample of the supervisor's response for the updating procedure for Telugu vowel recognition

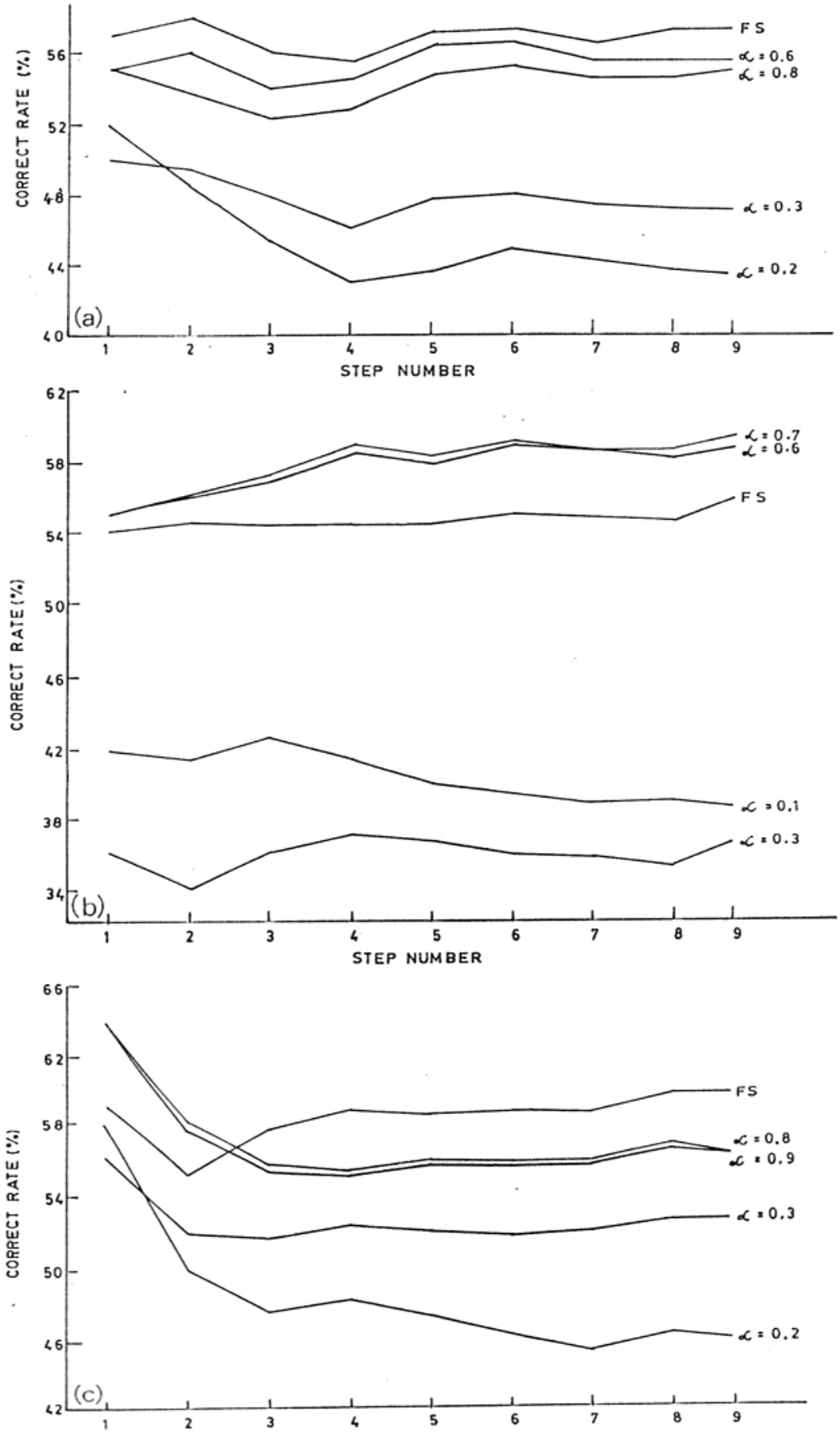

Fig.5.4 : System performance curves with Telugu vowel data when the initial estimates are 'very weak' and a Bayes classifier is used with λ-sequence 1 and the fully supervised case (for three different input sequences)

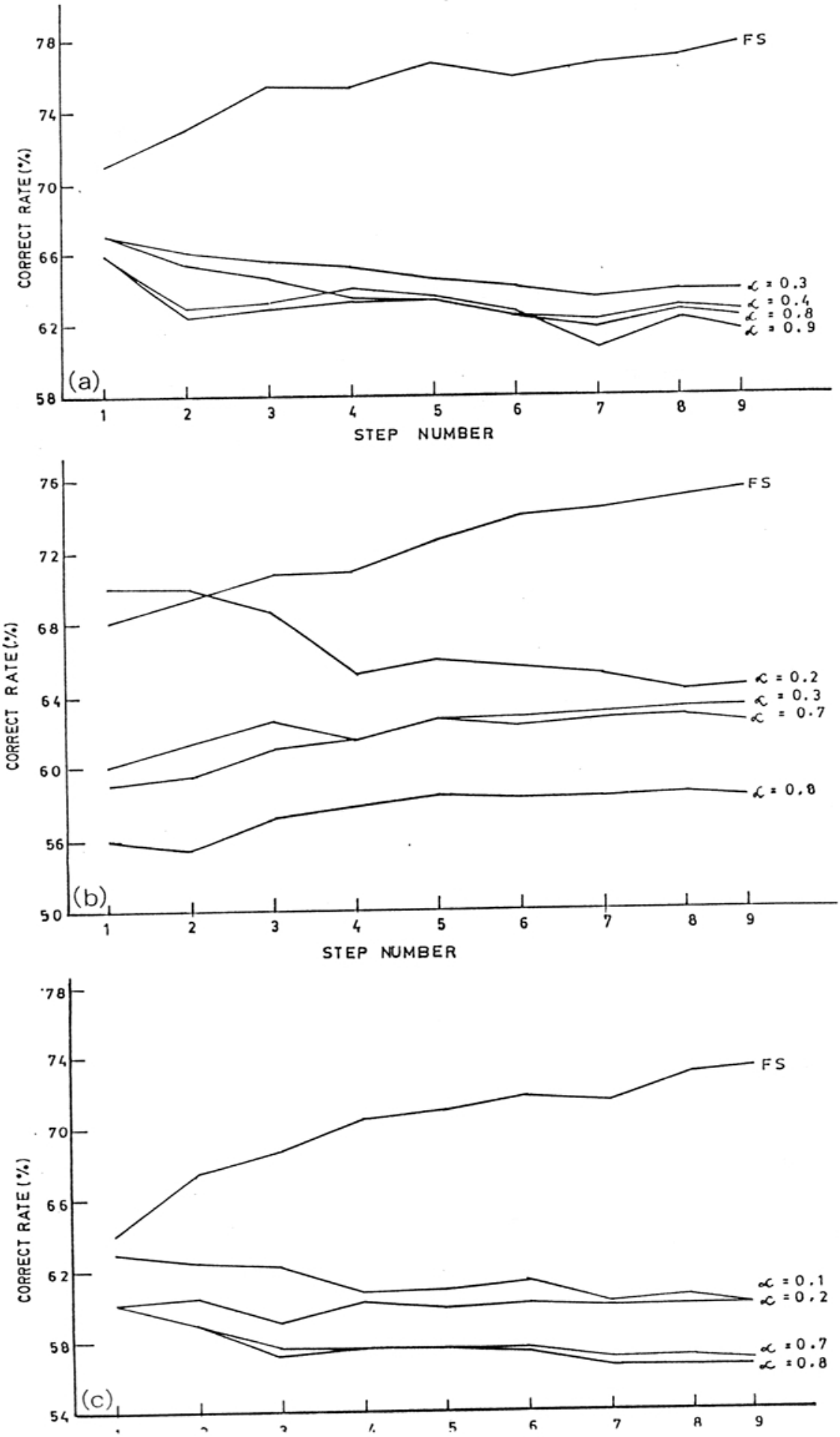

Fig.5.5 : System performance curves with Telugu vowel data when the initial estimates are 'not too weak' and a Bayes classifier is used with λ-sequence 1 and the fully supervised case (for three different input sequences)

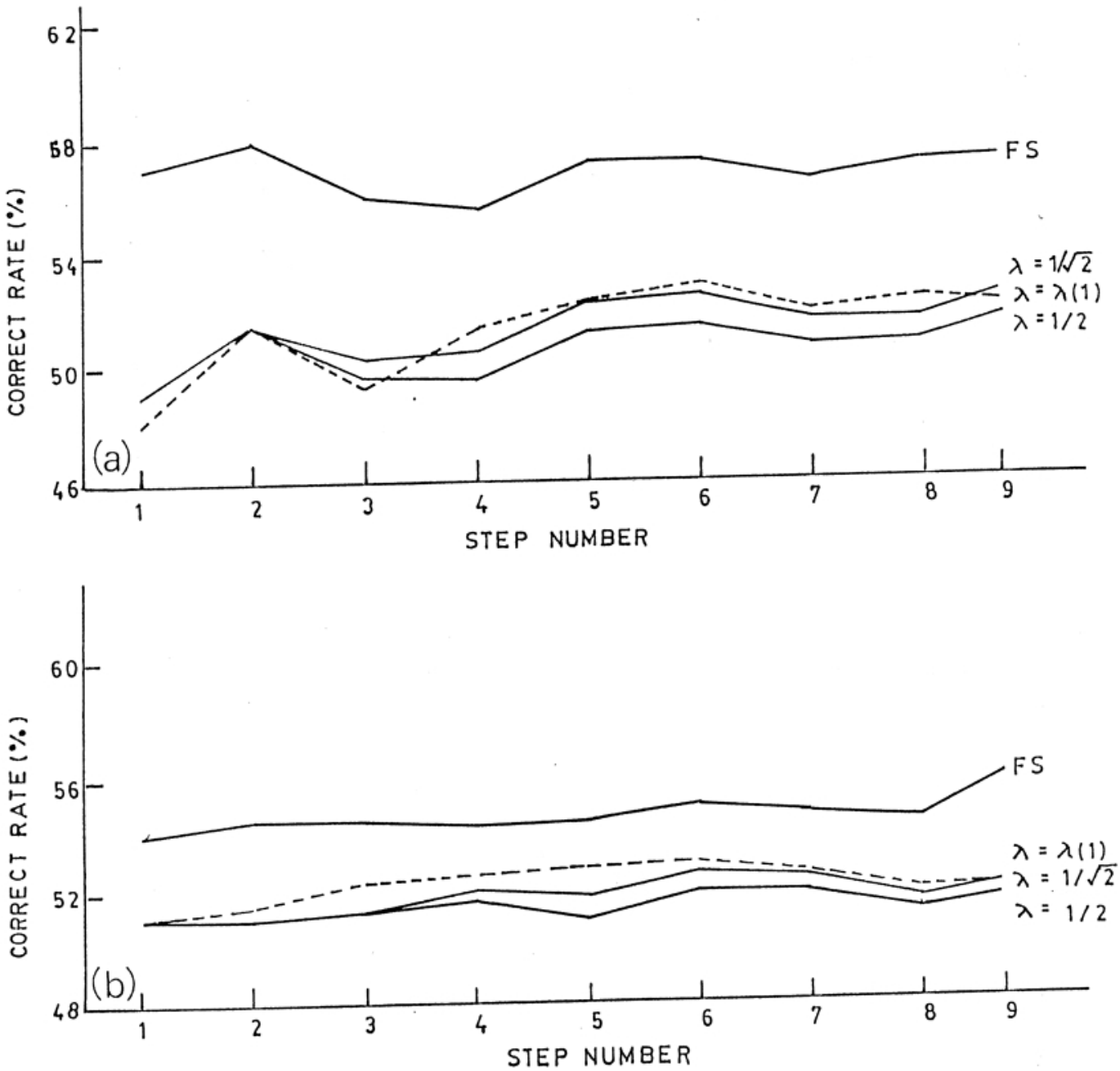

Fig.5.6 : System performance curves with Telugu vowel data when the initial estimates are 'very weak' and a Bayes classifier is used with fixed λ-values.

To demonstrate the truth of proposition 3.6, namely, that the GGA estimates manage, in the long run, to get closer to the true parameter values than the non-GGA under certain conditions, another experiment was conducted. Some more different orderings of the input sequence were used, with different initial estimates, based on 5% samples from the sample sets corresponding to each class, that is, with *not too weak* initial estimates. The GGA and the non-GGA were implemented and at each iteration the Euclidean distance of the estimates from the *true* values (computed with the help of the entire data set) was computed. The results are presented in the form of graphs in figures 5.7 and 5.8 [7], for two different orderings of the input sequence, the distances of the estimates from the true values being plotted at intervals of 50 iterations. Figure 5.7 corresponds to the mean vector and figure 5.8 gives the graph for the variance vector, covariances being assumed to be absent, at the very outset. From the figures 5.7 and 5.8, it is seen that, in the long run, the GGA estimates are closer to their true values than those obtained with the non-GGA.

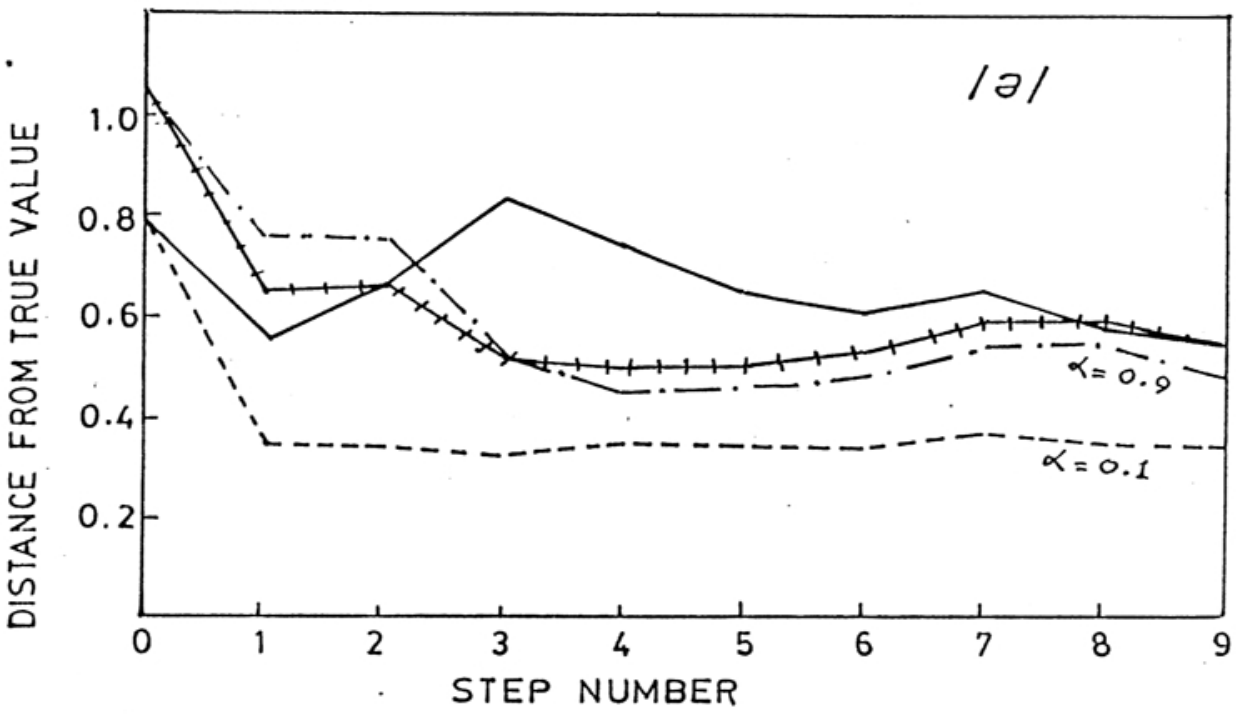

Fig. 5.7.1 : Distance of Estimated Mean Vectors from their True Values, for Telugu Vowel Recognition with Bayes classifier, for class /ə/

——— Non GGA with sequence 1

– – – GGA with sequence 1

+++++ Non GGA with sequence 2

— · — GGA with sequence 2

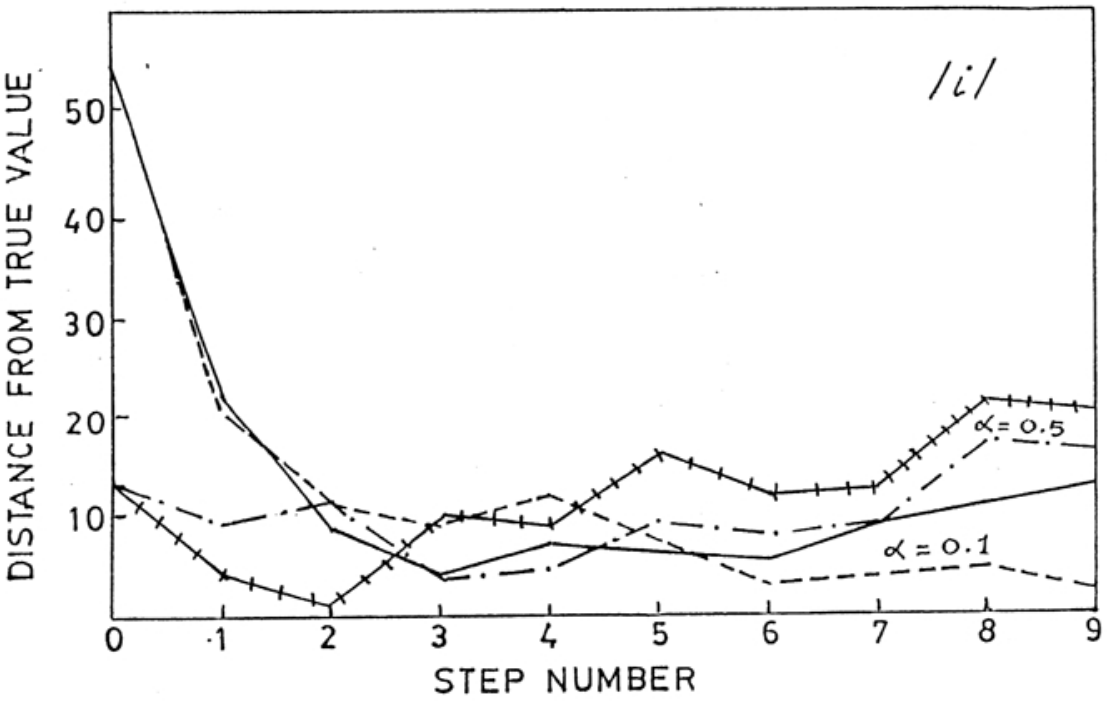

Fig.5.7.2 : Distance of Estimated Mean Vectors from their True Values, for Telugu Vowel Recognition with Bayes classifier, for class /i /

—— Non GGA with sequence 1

- - - GGA with sequence 1

++++ Non GGA with sequence 2

·—·— GGA with sequence 2

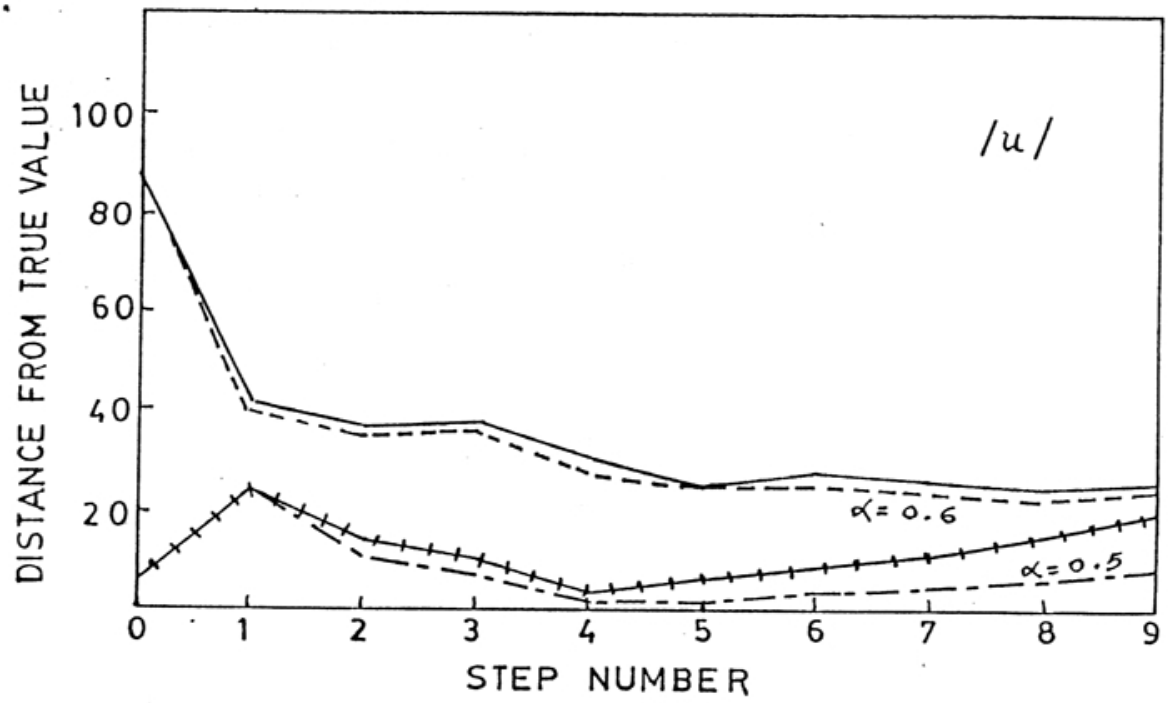

Fig. 5.7.3 : Distance of Estimated Mean Vectors from their True Values, for Telugu Vowel Recognition with Bayes classifier, for class /ఉ/

——— Non GGA with sequence 1

- - - GGA with sequence 1

++++ Non GGA with sequence 2

·—·— GGA with sequence 2

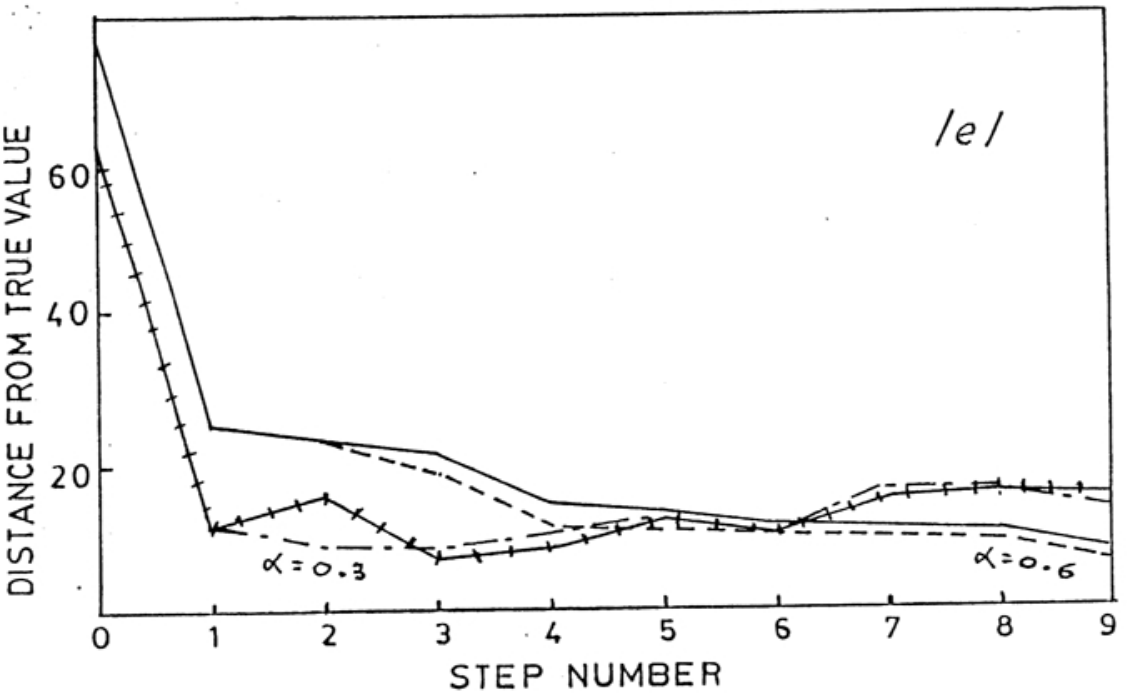

Fig. 5.7.4 : Distance of Estimated Mean Vectors from their True Values, for Telugu Vowel Recognition with Bayes classifier, for class /e/

—— Non GGA with sequence 1

– – – GGA with sequence 1

++++ Non GGA with sequence 2

–·— GGA with sequence 2

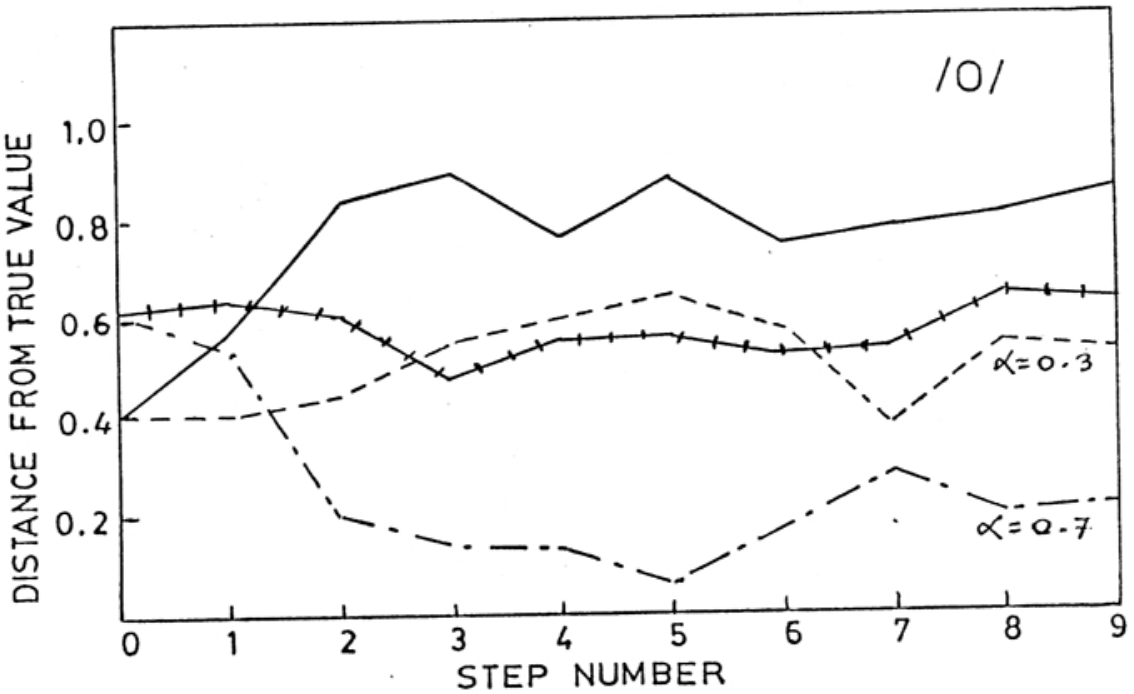

Fig. 5.7. 5 : Distance of Estimated Mean Vectors from their True Values, for Telugu Vowel Recognition with Bayes classifier, for class /O/

—— Non GGA with sequence 1

– – – GGA with sequence 1

++++ Non GGA with sequence 2

—·— GGA with sequence 2

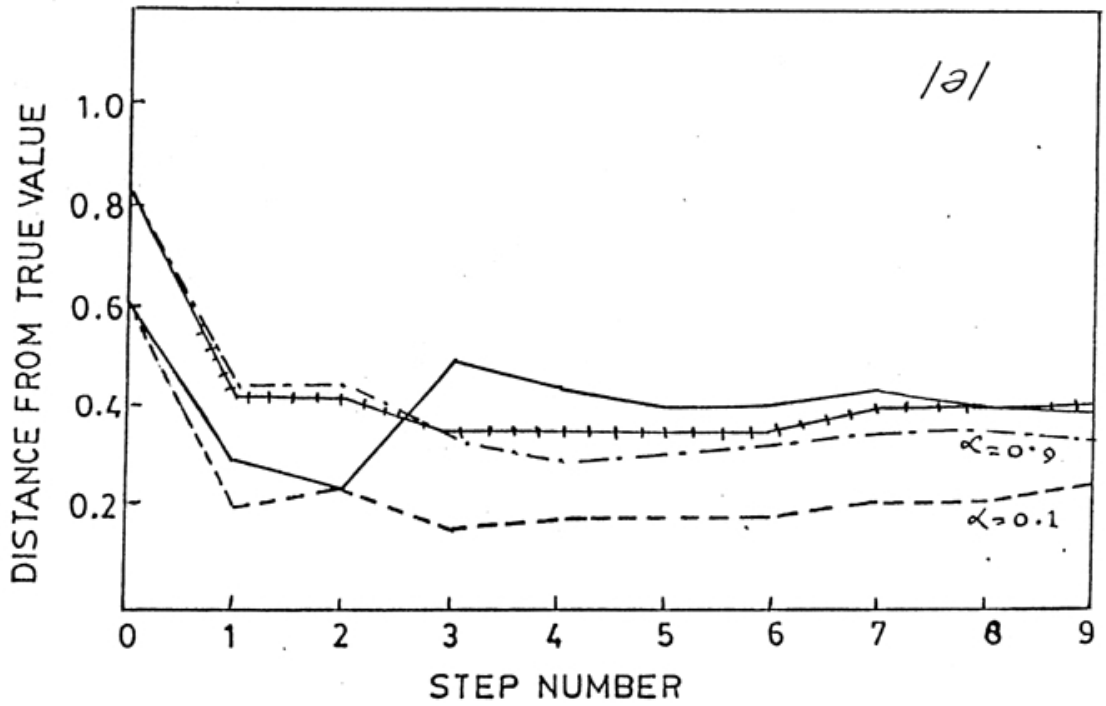

Fig. 5.8.1 : Distance of Estimated Variance Vectors from their True Values for Telugu vowel recognition with Bayes classifier, for class /ə/

—— Non GGA with sequence 1

– – – GGA with sequence 1

—— Non GGA with sequence 2

—·— GGA with sequence 2

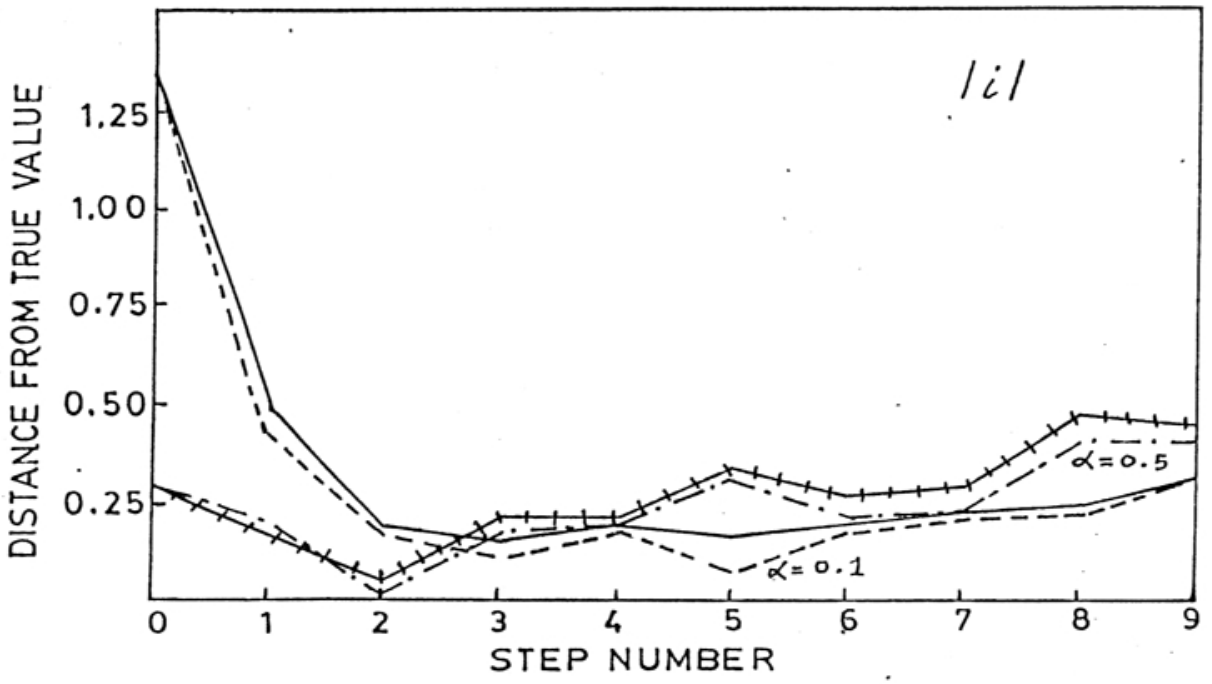

Fig.5.8.2 : Distance of Estimated Variance Vectors from their True Values for Telugu vowel recognition with Bayes classifier, for class / i /

—— Non GGA with sequence 1

– – – GGA with sequence 1

++++ Non GGA with sequence 2

—·— GGA with sequence 2

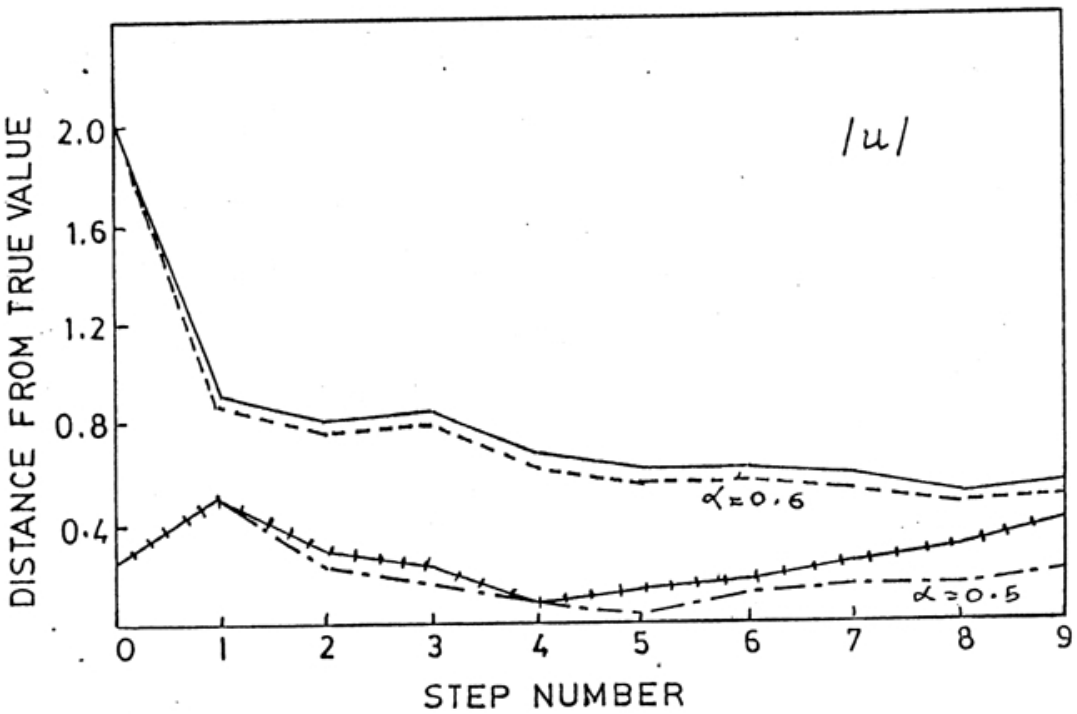

Fig. 5.8. 3 : Distance of Estimated Variance Vectors from their True Values for Telugu vowel recognition with Bayes classifier, for class /u/

—— Non GGA with sequence 1

- - - GGA with sequence 1

++++ Non GGA with sequence 2

—·— GGA with sequence 2

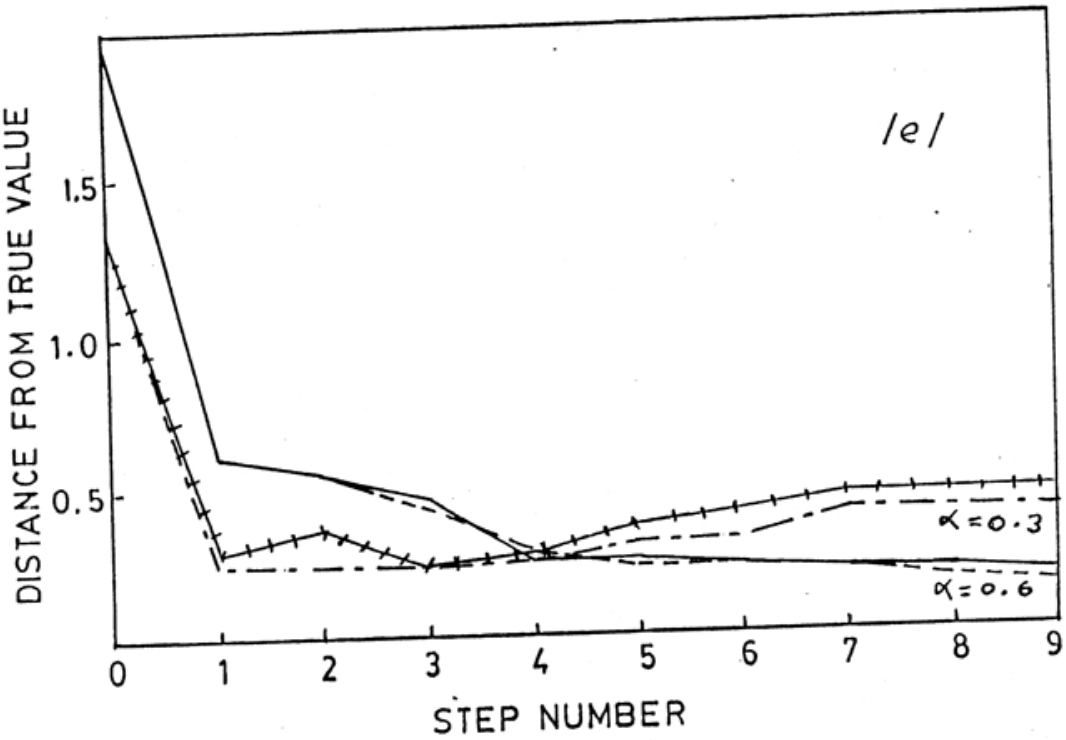

Fig.5.8.4 : Distance of Estimated Variance Vectors from their True Values for Telugu vowel recognition with Bayes classifier, for class /e/

——— Non GGA with sequence 1

- - - GGA with sequence 1

++++ Non GGA with sequence 2

—·— GGA with sequence 2

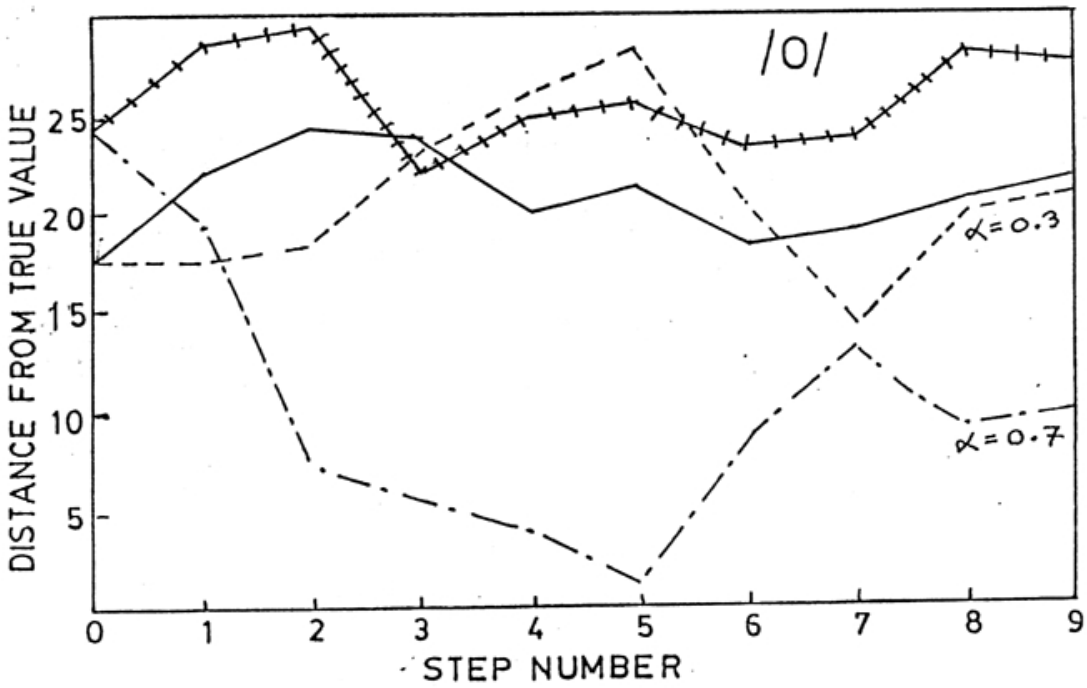

Fig.5.8.5 : Distance of Estimated Variance Vectors from their True Values for Telugu vowel recognition with Bayes classifier, for class / O/

—— Non GGA with sequence 1

— - — GGA with sequence 1

++++ Non GGA with sequence 2

—·— GGA with sequence 2

Part 2

Two cases are considered, depending upon the size of the training set used to obtain the initial estimate:

(i) the case of *very weak* initial estimates

(ii) the case of *not too weak* initial estimates

Using the Bayes classifier, the experiment is done exactly as for part 1 with four different input sequences. Cumulative % recognition scores are observed after every 100 samples are classified. The results are presented in a compact form in figures 5.9 and 5.10 for a single ordering of the input sequence. The first of the two figures corresponds to *very weak* initial estimates, and the second, to *not too weak* ones.

The following points were observed:

(i) In general, with respect to the criterion of % recognition score, the performance of the GGA with λ-sequence 1 is better than that of the non-GGA (identified by the label 'NS' in the figures, to emphasize the fact that it corresponds to the non-supervised scheme). Also, the performance of the GGA with λ-sequence 2 is generally comparable (at times, even better) than its performance with λ-sequence 1, in respect of the recognition score.

(ii) With *very weak* initial estimates, good GGA-performance with λ-sequence 1 is observed for higher values of α ($= 0.8, 0.9$) and with *not too weak* initial estimates, this is seen for lower values of $\alpha (= 0.1, 0.3)$.

(iii) In both the cases, the GGA with λ-sequence 1 gives a better recognition rate than the fully-supervised system (identified by the label 'FS' in the figures), for most situations.

(iv) With *not too weak* estimates, the performance of the GGA is far better than that of the non-GGA (that is, the non-supervised system), for both λ-sequences.

(v) With *very weak* initial estimates, the performance of the GGA is either better than or comparable to that of the non- supervised system, that is, the non-GGA, with both λ-sequences.

The results stated above for λ-sequence 1 are in agreement with results obtained in the first part and with those obtained earlier in [8].

5.3.3 For the terrain classification problem with LANDSAT data

The selfsupervised system (GGA) is compared with the non- supervised (i.e., the non-GGA) and fully-supervised systems in the same manner as in the Telugu vowel recognition problem.

As before, two cases are considered, depending upon the size of the training set used to obtain the initial estimate:

(i) the case of *very weak* initial estimates (3 samples)

(ii) the case of *not too weak* initial estimates (12% samples)

The cumulative recognition scores, mean vectors and covariance vectors are observed after every 100 samples are classified. The results are illustrated in figures 5.11, 5.12, 5.13 and 5.14 for two different orderings of the input sequence and for the two different sizes of the initial estimate that are mentioned above. The following points were noted:

(1) Good system performance of the GGA is observed with *very weak* initial estimates with $\alpha = .08, 0.9$, and with *not too weak* initial estimates with lower values of α, when λ-sequence 1 is used.

The above points were observed with Telugu vowel data also (see section 5.3.2 and [8].

(2) In all the cases, the performance of the GGA with both λ-sequences is better than that of the non-GGA (identified by the label 'NS' in the figures), in general.

Note

As a part of the above investigations, the nonsupervised, fully-supervised and non-adaptive (that is, where no updating of parameters takes place) schemes were also implemented for a large number of orderings of the input sequence and for different sizes of the training sample set used to obtain the initial estimates. The average values (computed over all the orderings) of the % recognition scores were plotted against the (initial) training sample sizes. The resulting graphs (given in figure 5.15) confirm something that can be felt intuitively, namely, that if the initial samples are good enough, there are hardly any differences among the performances of the three schemes.

Similar results with Telugu vowel data are available in [87,89,53].

The above findings serve to clarify why we had confined ourselves to very weak/not too weak initial estimates and not stronger ones, while demonstrating the adaptive ability of the learning system in all the experiments described earlier.

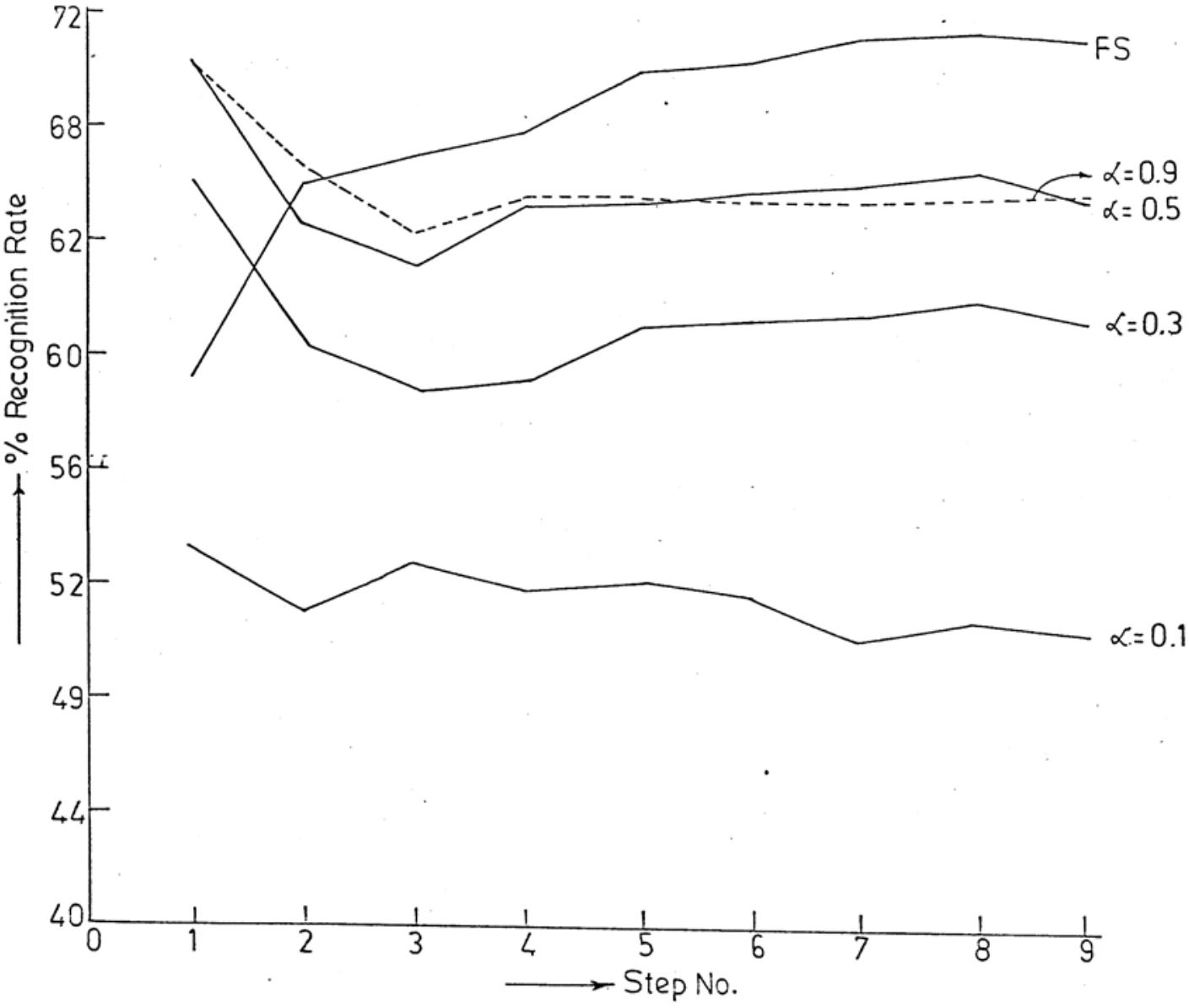

Fig.5.9.1 : System performance curves with Telugu vowel data when the initial estimation are 'very weak' and a Bayes classifier is used with -sequence 1 and in the fully supervised case.

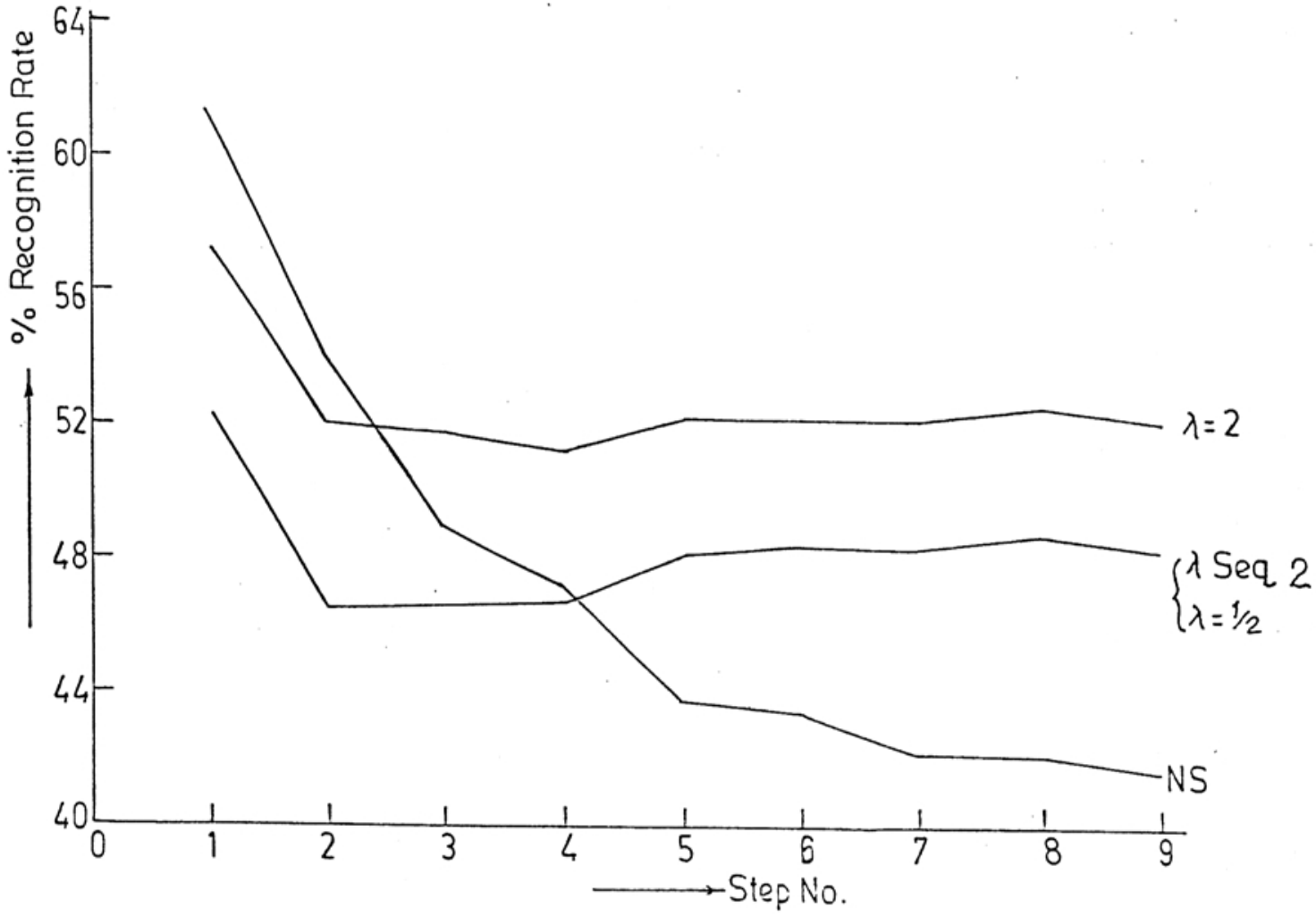

Fig.5.9.2 : System performance curves with Telugu vowel data when the initial estimates are 'very weak' and a Bayes classifier is used with -sequence 2, in the non-supervised case and with fixed values (=1/2,2).

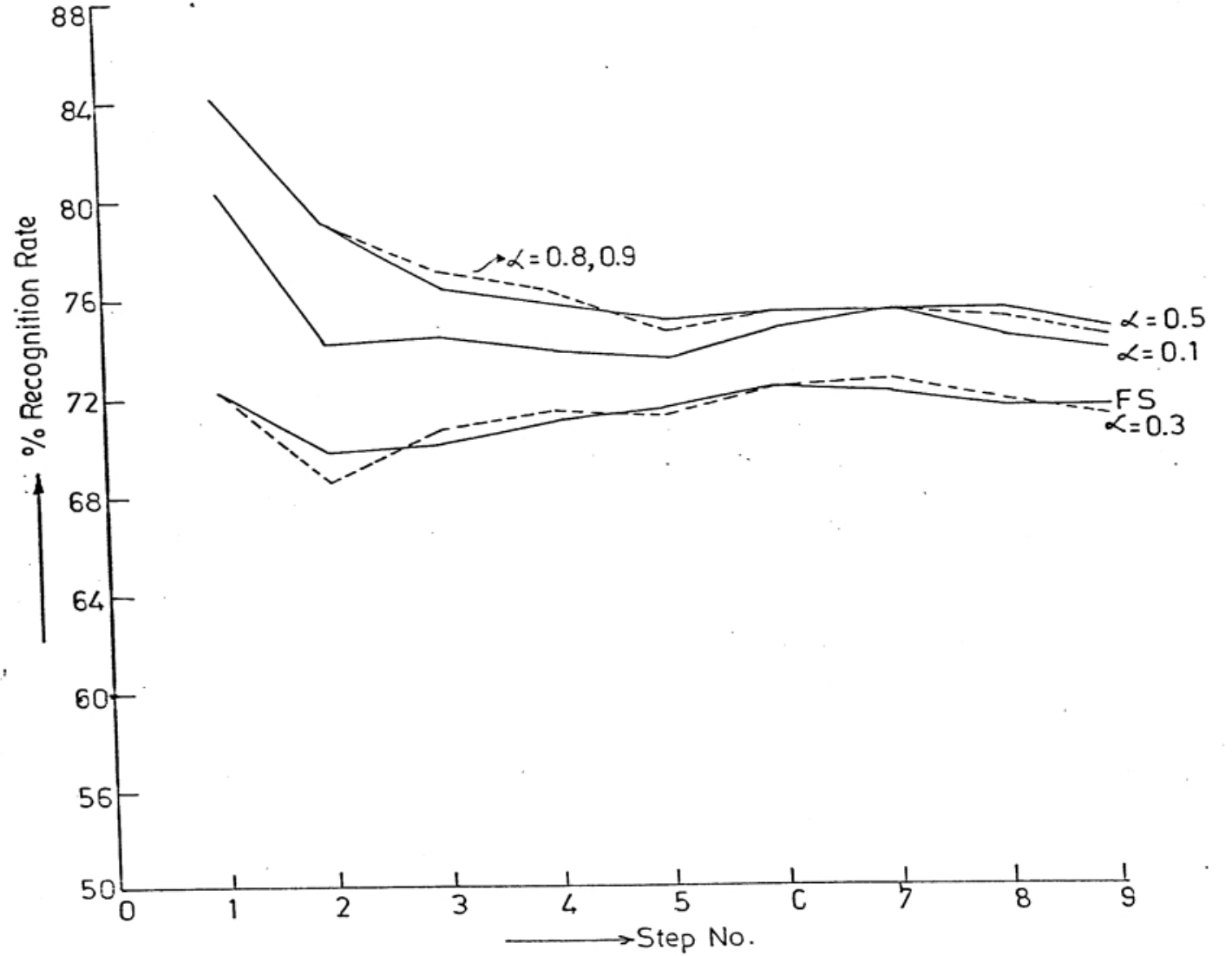

Fig.5.10.1 : System performance curves with Telugu vowel data when the initial estimates are 'not too weak' and a Bayes classifier is used with λ-sequence 1 and in the fully supervised case.

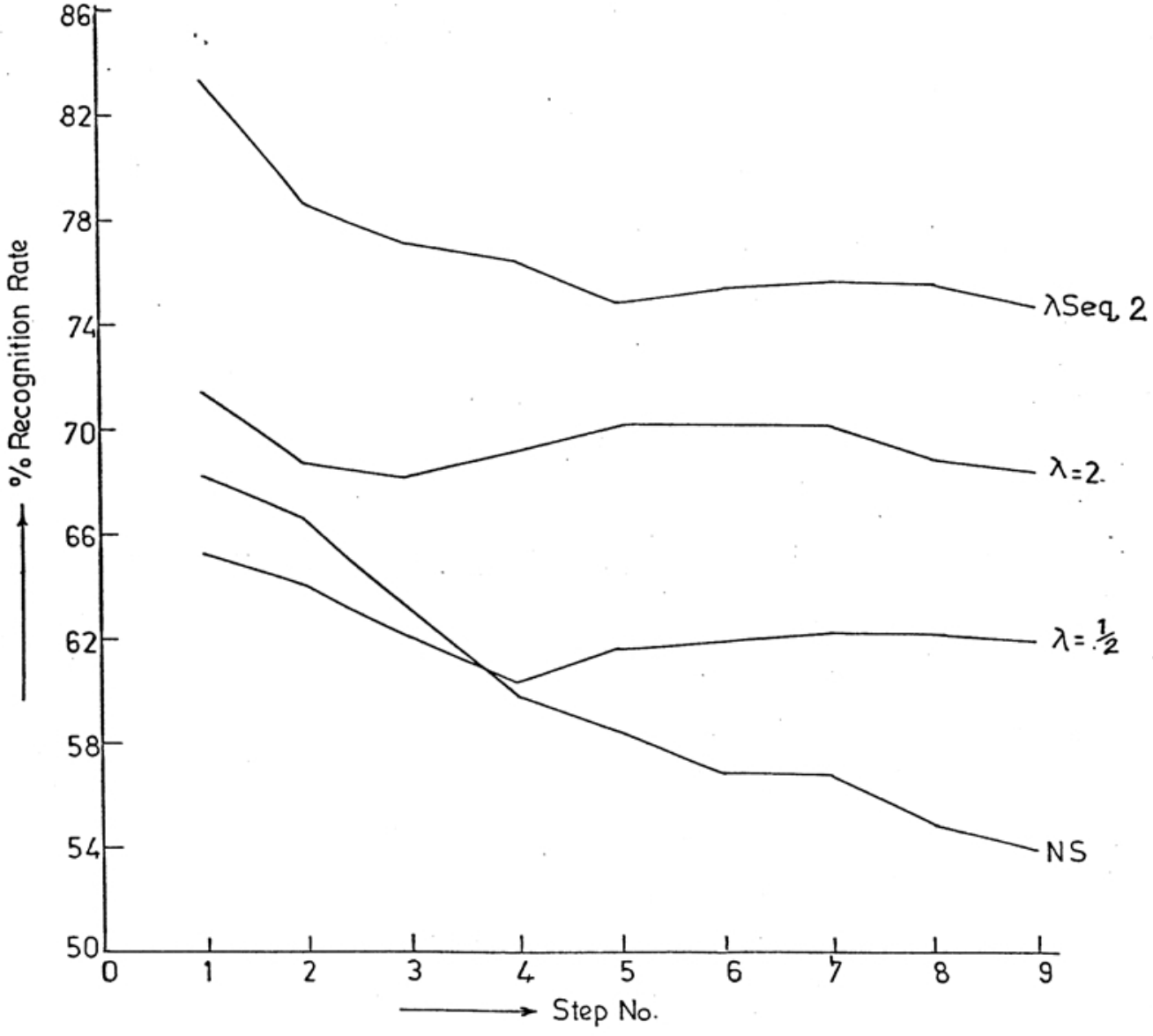

Fig.5.10.2 : System performance curves with Telugu vowel data when the initial estimates are 'not too weak' and a Bayes classifier is used with λ-sequence 2, in the non-supervised case and with fixed values (=1/2,2).

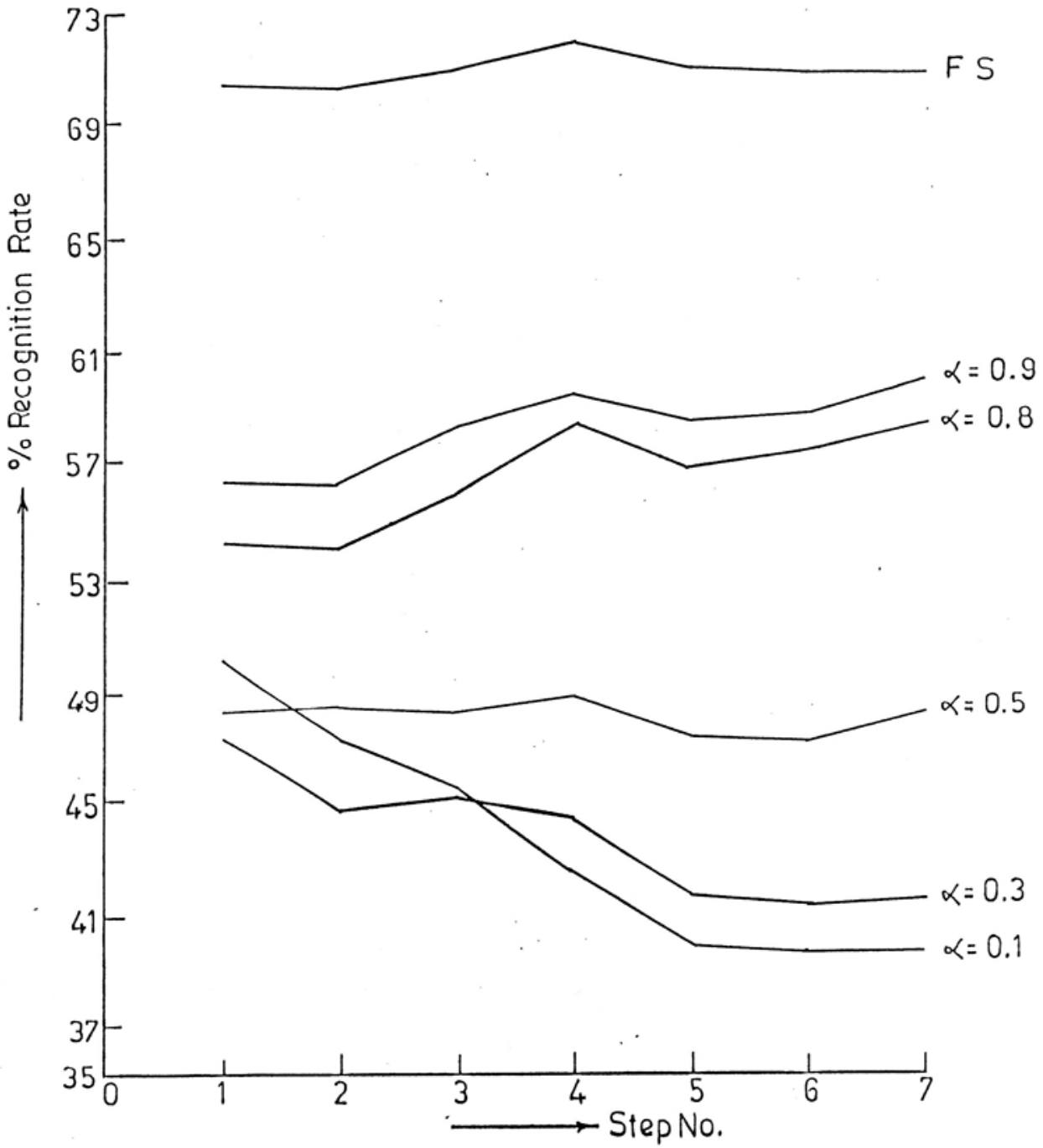

Fig.5.11.1 : System performance curves with LANDSAT(MSS) data when the initial estimates are 'very weak' and Bayes classifier is used with λ-sequence 1 and in the fully supervised case, with input sequence 1.

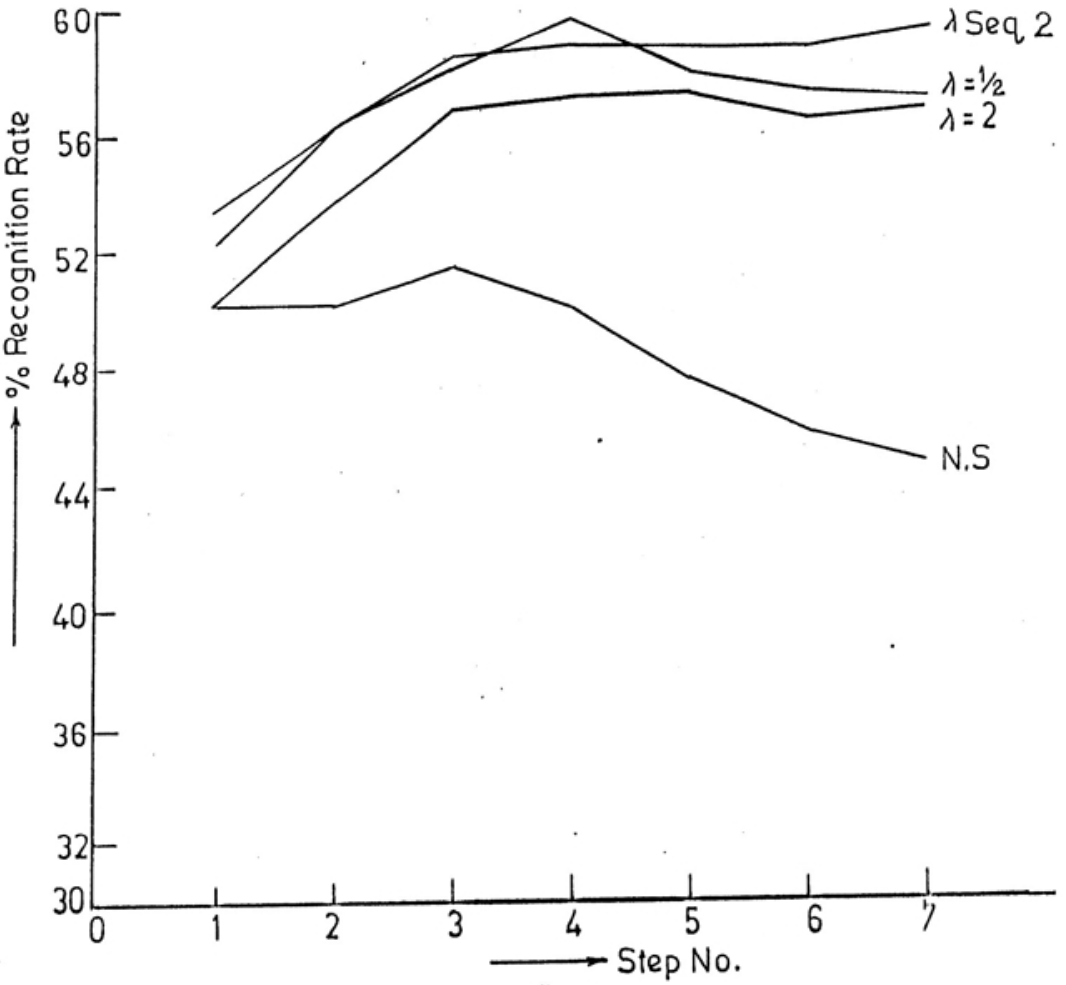

Fig.5.11.2 : System performance curves with LANDSAT (MSS) data when the initial estimates are 'very weak' and a Bayes classifier is used with λ-sequence 2, in the non-supervised case and with fixed values(=1/2,2), with input sequence 1.

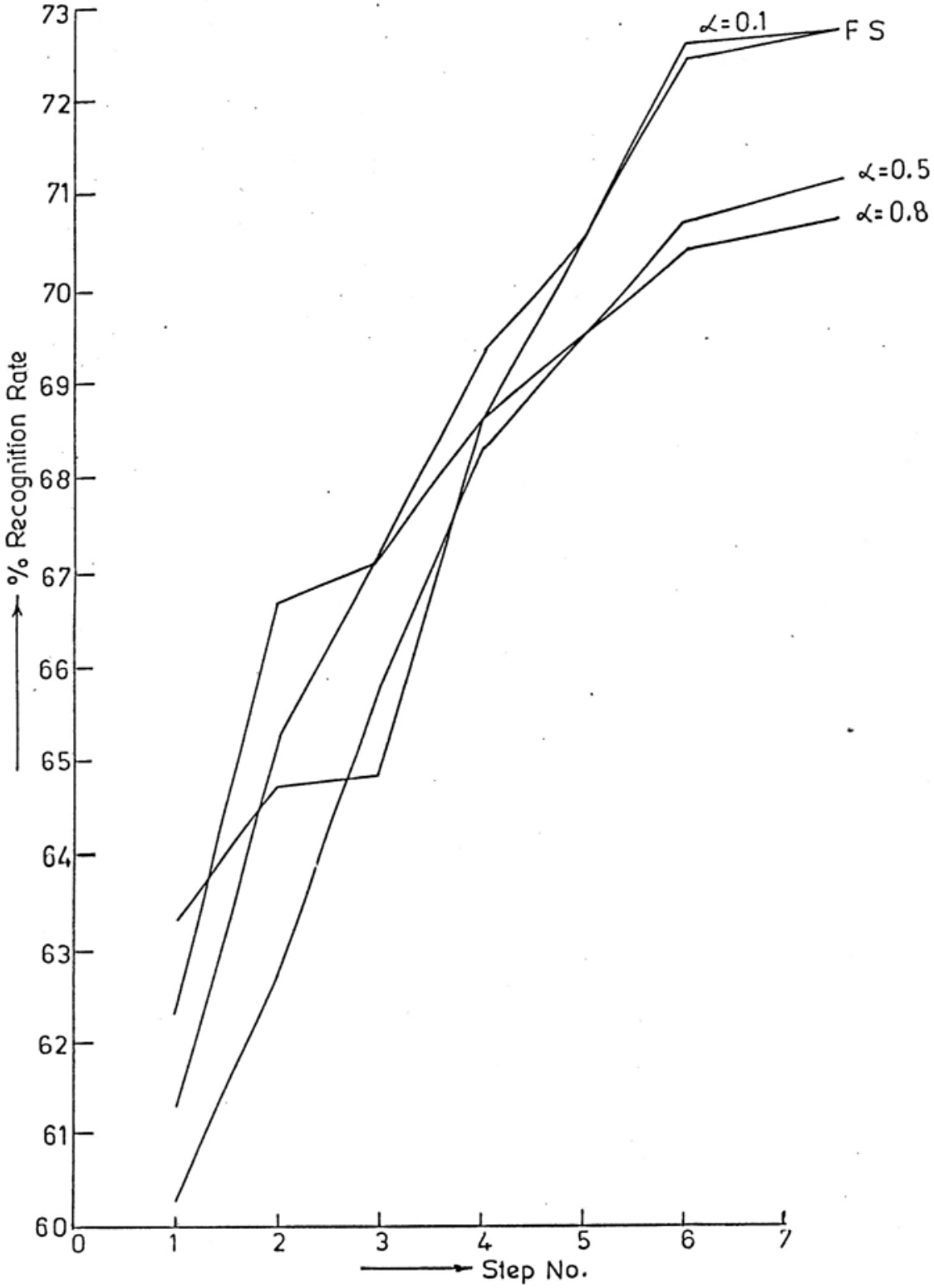

Fig. 5.12.1 : System performance curves with LANDSAT (MSS) data when the initial estimates are 'not too weak' and Bayes classifier is used with λ-sequence 1 and in the fully supervised case, with input sequence 1.

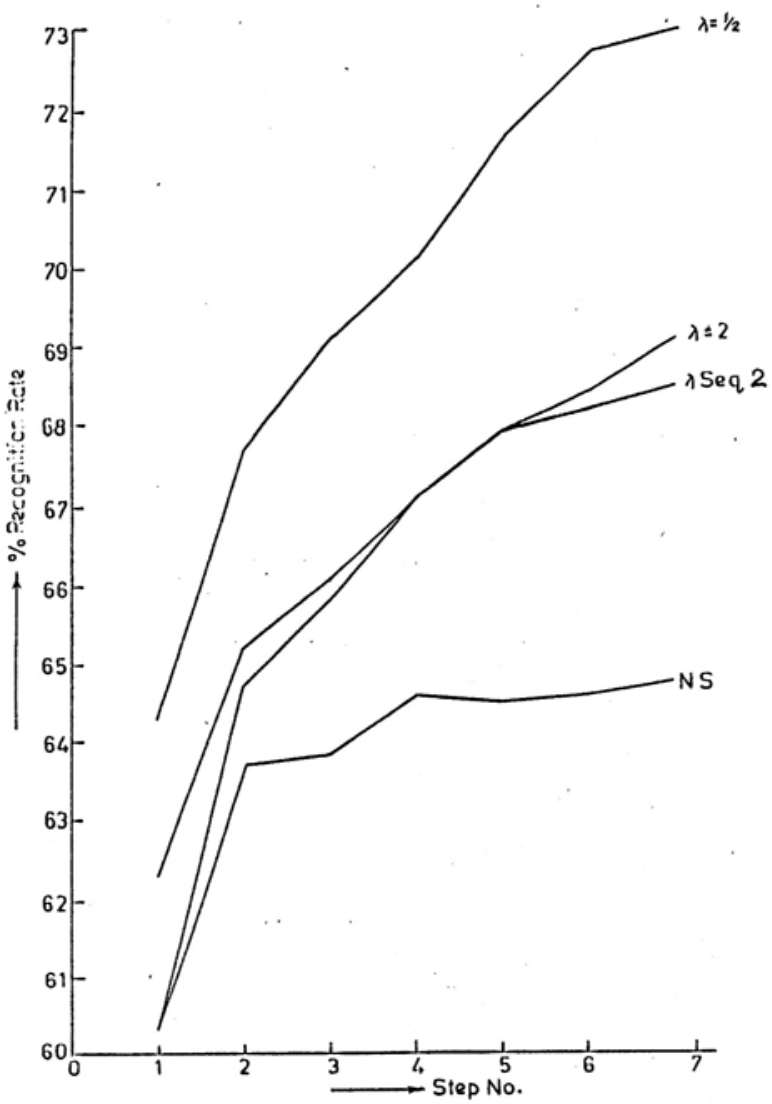

Fig. 5.12.2 : System performance curves with LANDSAT (MSS) data when the initial estimates are 'not too weak' and a Bayes classifier is used with λ-sequence 2, in the non-supervised case and with fixed values (=1/2,2), with input sequence 1.

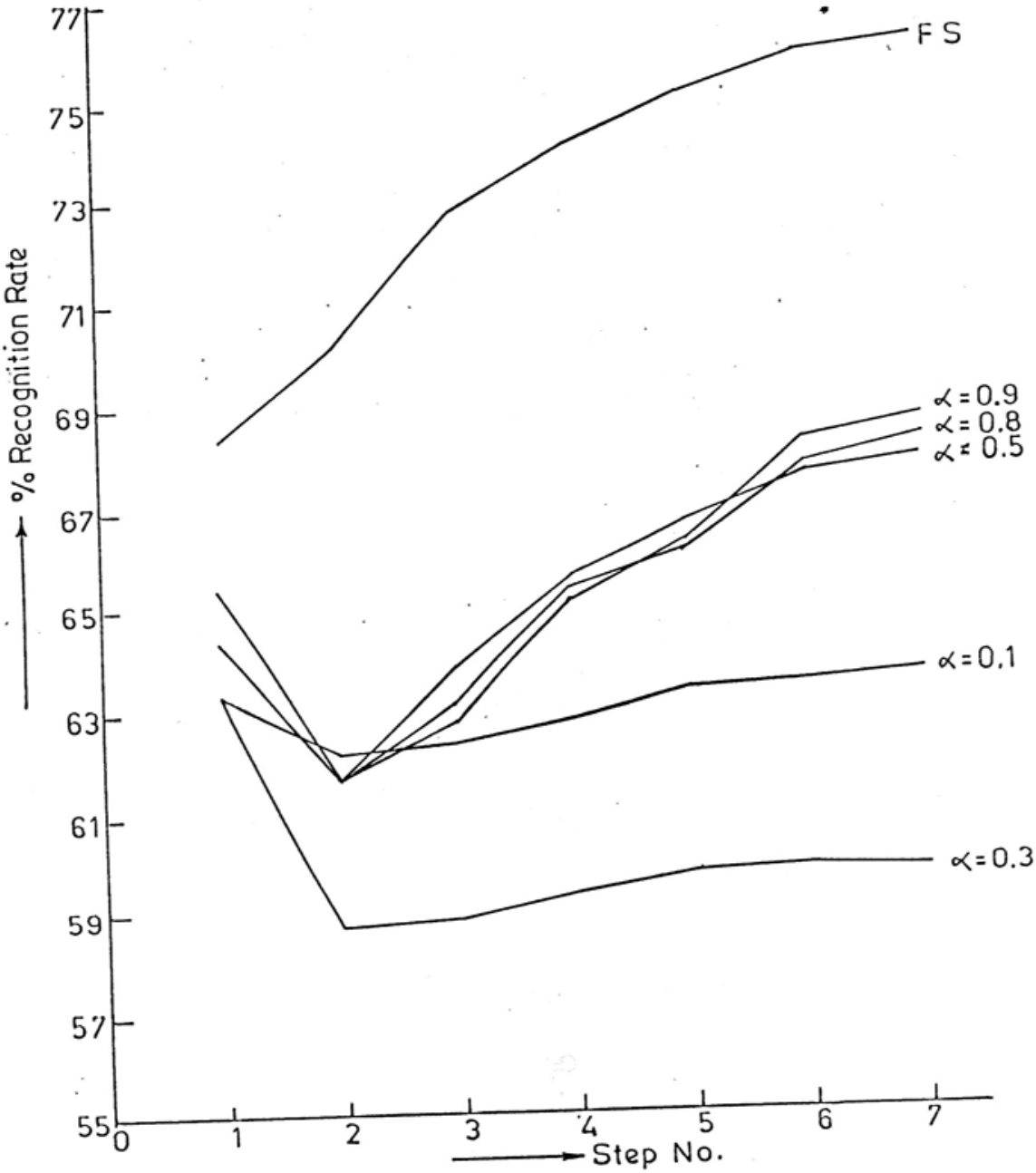

Fig. 5.13.1 : System performance curves with LANDSAT (MSS) data when the initial estimates are 'very weak' and Bayes classifier is used with λ-sequence 1 and in the fully supervised case, with input sequence 2.

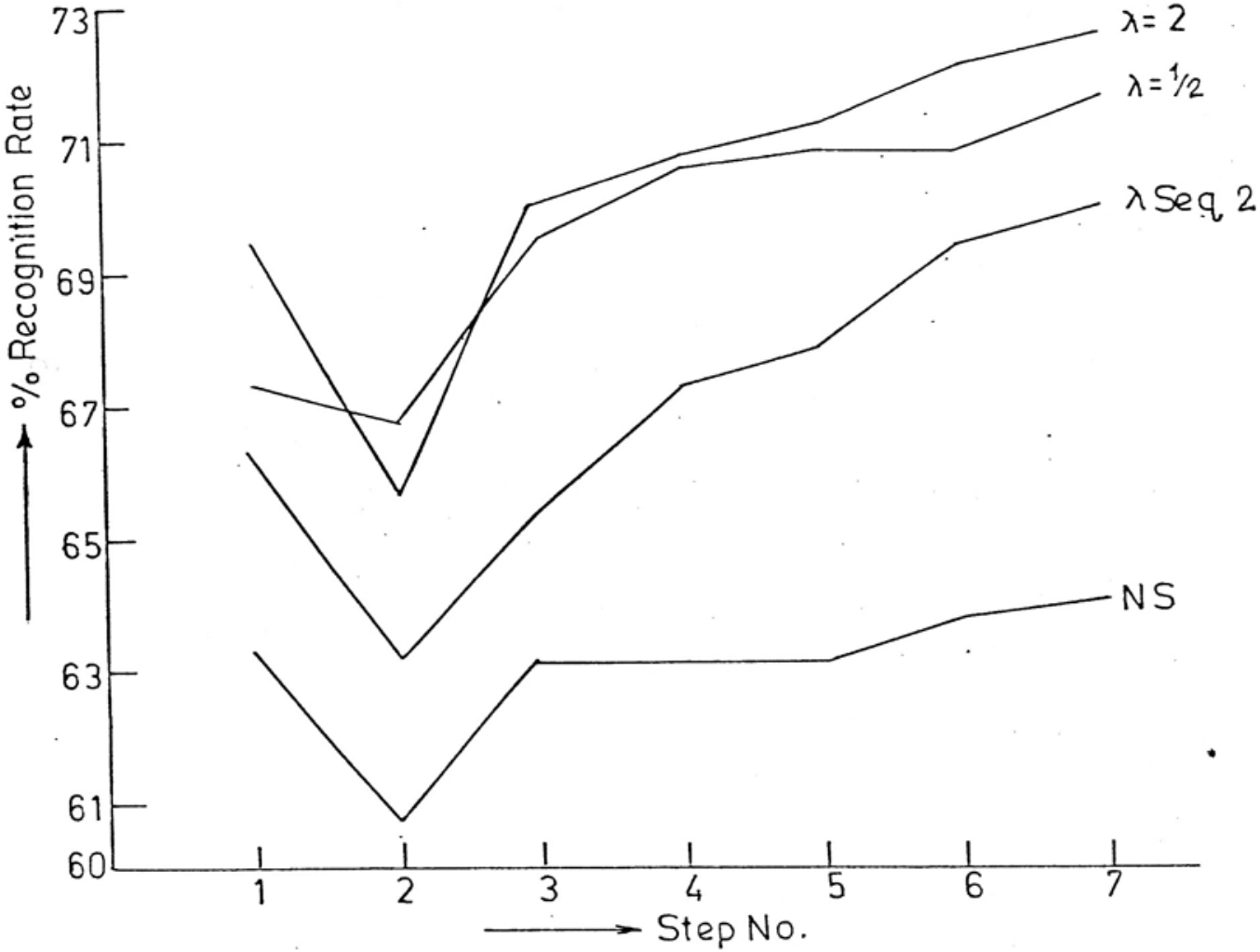

Fig. 5.13.2 : System performance curves with LANDSAT(MSS) data when the initial estimates are 'very weak' and a Bayes classifier is used with λ-sequence 2, in the non-supervised case and with fixed values (=1/2,2), with input sequence 2.

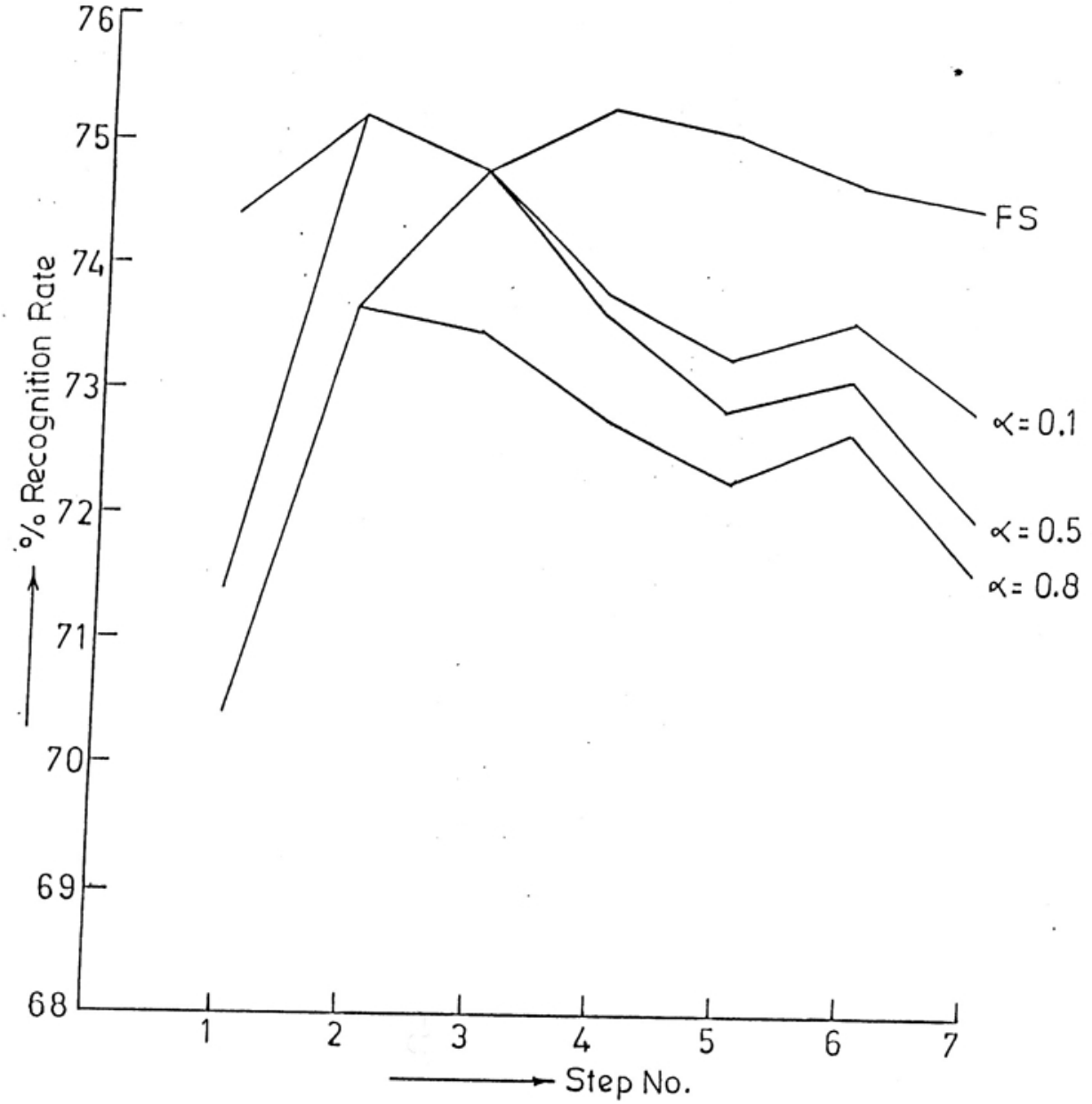

Fig. 5.14.1 : System performance curves with LANDSAT (MSS) data when the initial estimates are 'not too weak' and Bayes classifier is used with λ-sequence 1 and in the fully supervised case, with input sequence 2.

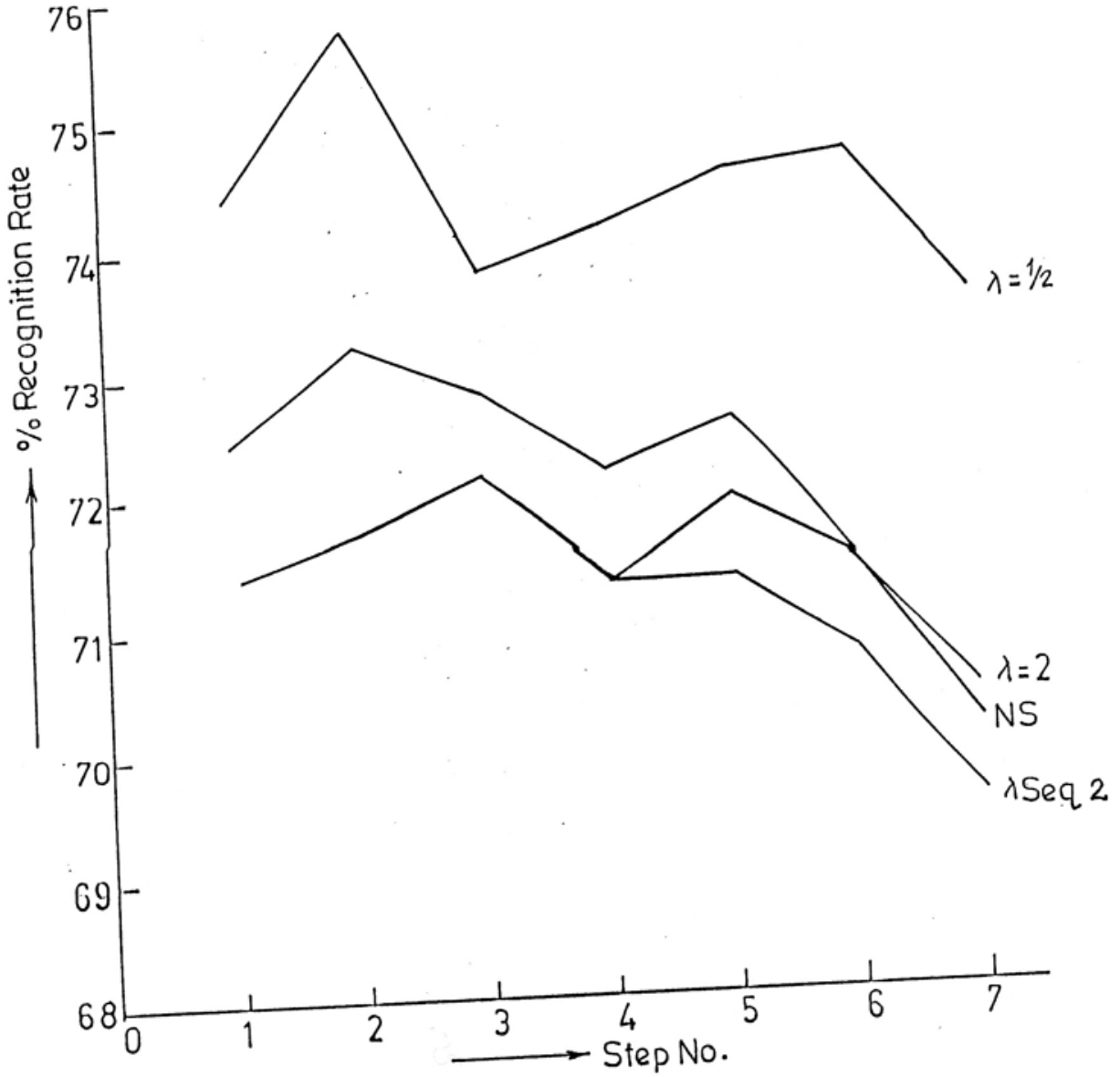

Fig.5.14.2 : System performance curves with LANDSAT (MSS) data when the initial estimates are 'not too weak' and a Bayes classifier is used with λ-sequence 2, in the non-supervised case and with fixed values (=1/2,2), with input sequence 2.

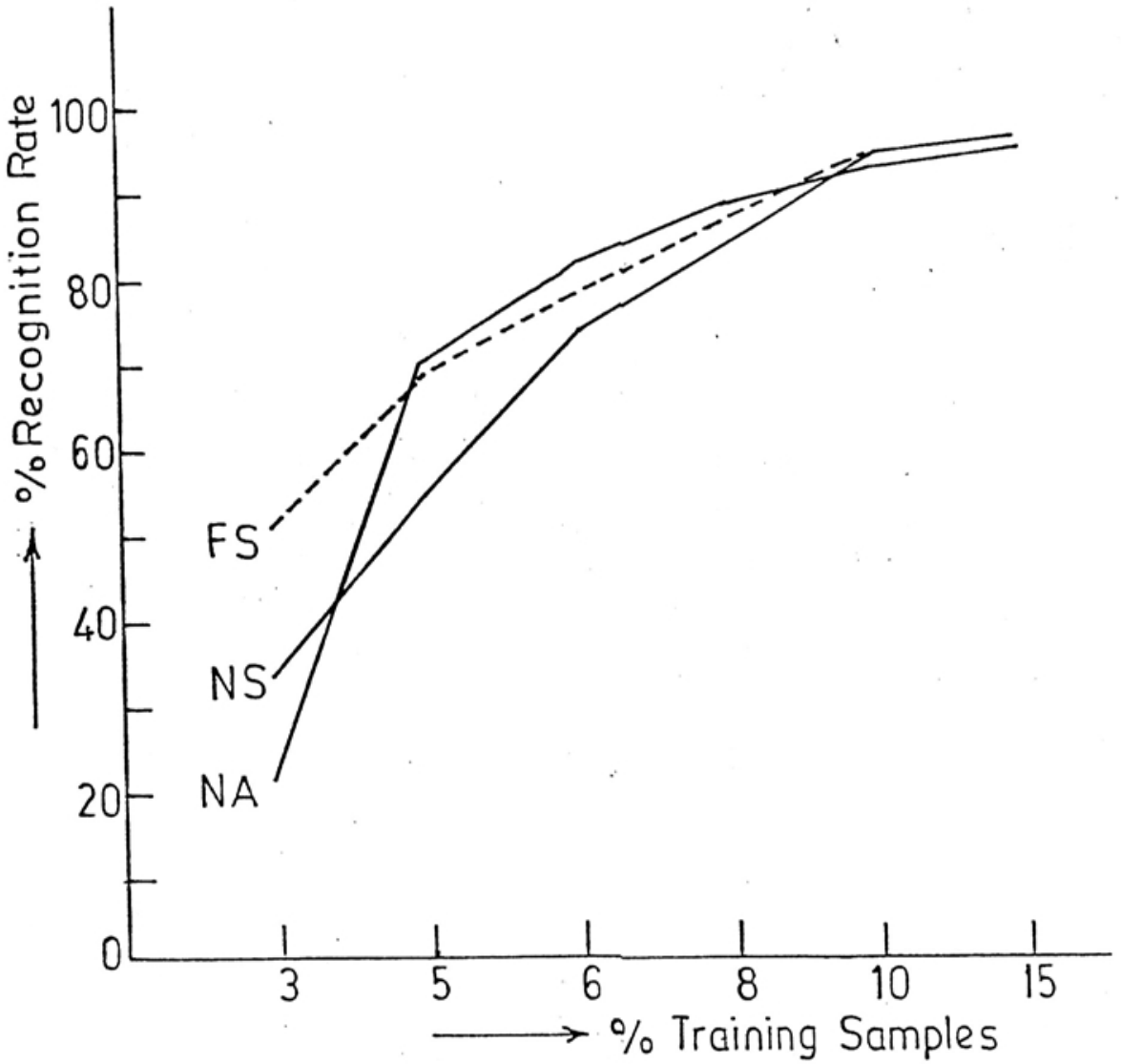

Fig.5.15 : Variation of 'average' recognition rate with size of initial training set, with the fully-supervised(FS), nonsupervised (NS) and non-adaptive(NA) learning schemes, for LANDSAT(MSS) data.

Chapter 6

Conclusions and suggestions for further work

6.1 Summary of contributions

In this thesis, certain aspects of a class of parameter learning algorithms for pattern recognition, based on the stochastic approximation approach to nonparametric learning, have been investigated. The main feature of this class of algorithms is that it is designed to learn parameters recursively in an error-prone environment where there is a possibility of unreliable or nonrepresentative training samples being available. The algorithms under study aim to detect and weed out such samples in order to prevent the estimates from being corrupted by them. These algorithms have been labeled Generalised Guard Zone Algorithms (GGAs, in short) in this work, since they are actually generalised versions of certain existing algorithms of this type, due to Chien and Pal et al. [54,90,53]. The term 'guard zone' is there to remind us that basically all these algorithms are based on the concept of a hyperellipsoid of certain dimensions (that are functions of a parameter λ), centered at the past estimate of the mean. The guard zone acts as supervisor by rejecting or accepting the current training sample for updating purposes, by examining whether it (the sample) lies within it or not. Being subjective rules for the rejections of outliers, such algorithms can be considered to be robust statistical procedures.

This work tries to answer some of the questions left unanswered, or only partially answered by Chien and Pal et al., for instance, about the large-sample (asymptotic) behaviour of such algorithms, and the proper choice of the parameter λ. This work also makes the theory more relevant to practical situations by doing all investigations under a probability model (originally due to Chittineni [77]) which takes into consideration the possibility of wrong labeling of training samples. The general m-class N-feature pattern recognition problem is considered. The findings are demonstrated with the help of an artificial data set, a data set for Telugu vowel recognition, and a data set for terrain classification derived from LANDSAT (MSS) data.

At first, a mathematical description of the GGA is formulated, and its relation with the existing algorithms is described in full. At the next step, the stochastic convergence of the GGA is investigated in the ideal situation where there is no mislabeling of training samples. As expected, the GGA converges with probability 1 and in the mean square to the true parameter value, like other non-guard-zone-based stochas-

tic approximation algorithms (which have been called non-GGA for ease of reference), under certain conditions.

At this stage, the model for mislabeling is brought into the picture. Under this model, it is established that the non-GGA converges with probability 1 and in the mean square, but to non-true values, specifically, to certain convex linear combinations of the true parameter values for all m classes, the combining coefficients being simple functions of the *a priori* probabilities of the classes and the mislabeling probabilities. Further, it is also established that under similar conditions, the GGA too, converges with non-true values which are linear (not necessarily convex) combinations of the true parameter values of all the classes. The combining coefficients in this case involve, obviously, a certain conditional probability of inclusion in the guard zone. These results, though expected, are important in the sense that they *quantify* the effect of the presence of mislabeled samples on the asymptotic behaviour of the guard zone. Next, a proposition is proved, which establishes that for a certain *relative* configuration of the m classes in the feature space, the GGA is able to get closer (in the sense of Euclidean distance) to the true value for that class than the non-GGA does.

In another part of the work, an upper and a lower bound to the parameter λ of the GGA is derived, and a convex linear combination of the two, with combining coefficient α is proposed as an estimate for λ. Pal et al. and Chien had assumed constant values of λ in their experiments although the algorithm of Chien theoretically involves the notion of a dynamic threshold. We argued, however, that the guard zone would be a more sensitive supervisor if it is not kept fixed at a certain constant value, but is allowed to depend in some way on the training sample. Since the bounds obtained here are functions of the training sample, the estimate of λ based on them satisfies this requirement. The application of the GGA to various sets of data with this estimate of λ, for different values of α, demonstrated admirably the conjecture that the effect of making the guard zone dynamic in the sense of making λ variable, is to improve the performance of the learning PR system. The results are presented in this thesis, for the Telugu vowel recognition problem, and for terrain classification with LANDSAT data. A Bayes classifier was used so that the mean vectors and the covariance matrices were the parameters to be learned. It was also noted that the appropriate value of α to use depends on the nature of the initial sample. Generally, α is required to be larger for poorer initial estimates and smaller for better ones. This is consistent with the intuitive view that if the initial estimate is poor, the guard zone supervisor needs to be lenient (that is, λ should be large) in order to allow new samples to enter and (possibly) improve the estimates.

In the final part of the thesis, another method for obtaining estimates for λ is proposed, which is based on the minimum mean squared error (MSE) approach. At first, the closed form expression for the MSE is obtained which is minimised with respect to p_n, the probability of inclusion in the guard zone with parameter λ_n, n being the iteration number. It is found that the MSE will be minimum if p_n is a certain function of a_n and a_{n+1}, where the a_i's are generally prespecified parameters of the GGA which must satisfy the condition

$$\sum_{i=1}^{\infty} a_i^2 < \infty$$

in order that the GGA may converge. It is assumed that

$$a_n = 1/n$$

and some results from large-sample statistical distribution theory are applied, to obtain *distribution-free* estimates for λ_n for $n > N$. These are functions of N and n only, and involve the percentage points of the beta-variate with degrees of freedom dependent on N and n. These estimates too, are dynamic and experimental results obtained with all three data sets show that the performance of the GGA with these values of λ is better generally, as compared to the non-GGA. These results too, are presented in this thesis.

6.2 Suggestions for further research

As mentioned in sec 2.5, the GGA qualifies as a robust statistical procedure on the grounds that it is a type of subjective rejection rule for outliers. It was also noted there that as such, it is a rather unsophisticated robust procedure. The theory of robust statistical methods admits of a number of more sophisticated estimation procedures [56,57,59,58] which have not been tried out by pattern recognition scientists for learning purposes. However, most of these techniques are non-recursive in nature. In fact, the theory of recursive robust estimates, specifically for multidimensional parameter vectors, is not too developed [59]. Therefore, one direction in which further research can be aimed at, is towards exploiting the theory of robust statistics, extending and modifying existing theory if necessary, to get possibly better techniques for recursive estimation of parameters. For instance, it would be interesting to see whether the use of an existing robust statistic (or some modification of it) in place of similar estimates being used in PR can prove to be useful in terms of saving in computer time/space or improvement in the performance of the classifiers based on them. This kind of investigation can either be attempted theoretically or can be of an empirical nature. One obvious advantage of robust statistical procedures is that they are relatively insensitive to small variations in the underlying assumptions, and as such, can be expected to perform better in many types of non-ideal situations, for example, the type that arises due to the presence of mislabeled samples. Therefore, any fruitful research in this direction might prove to be quite beneficial to the field of pattern recognition in particular, and learning theory in general. It might also lead to new results in the theory of robust statistical methods as well.

Returning to the GGA, we recapitulate that in this thesis two methods have been proposed for estimating the parameter λ — one depending on certain upper and lower bounds to it, and the other based on the technique of minimizing the mean squared error (MSE). It might prove fruitful to investigate whether more sophisticated techniques like stochastic programming and related optimization methods can be adapted for this particular problem, that is, the estimation of λ in a manner which is efficient with respect to some criterion (of efficiency). For instance, if some statistical classifier is to be used, the criterion could be to minimize the probability of misclassification. Actually, instead of considering λ directly, one can try optimizing the value of α, where α is as in section 3.7, that is,

$$\alpha = \frac{\lambda - \ell}{L - \ell},$$

where ℓ and L are respectively the lower and upper bounds of λ and are functions of the training samples. It is difficult to say offhand which of the approaches will be more complicated.

Albert and Gardner [49] have proposed certain nonlinear stochastic approximation algorithms which involve truncated values of the observations. It would be interesting to see how this approach can be adapted to our kind of problem, and to see whether it is better to discard the unreliable sample altogether, as the GGA does, or to truncate it and then use it for updating. The main problem here is of course, the truncating rule, and suitable criteria will have to be selected for designing it.

Also, as fuzzy set theory is ideally suited for describing vague or imprecise situations, it might be worthwhile to examine whether it can be used to model the situation that results from the presence of unreliable samples, and to develop algorithms similar to the GGA. Incidentally, some work has been done in the direction, for instance, by Stephanou [91], who has suggested an imperfectly supervised (a hybrid supervised/unsupervised) scheme based on fuzzy prototypes. This scheme can be used when training samples are unreliable or very few in number.

Another relevant line of research might be to investigate whether it is possible to develop an interactive learning PR system [92] based on the GGA, which can interactively utilize some knowledge-base, for instance, that of the system user, in order to improve its performance.

Finally, it might also be interesting to see that if estimates for the mislabeling probabilities are available, for instance, in terms of the misclassification probabilities in case a statistical classifier is used, whether better estimates for λ or even, better learning rules can be obtained based on them. In general, the estimation of the mislabeling probabilities can be an important subproblem of this. Some research has already been done in this direction, notably, by Chittineni [77,62].

Bibliography

[1] P. W. Becker, *Recognition of Patterns.* Wien: Springer–Verlag, 1978.

[2] L. N. Kanal, *Pattern Recognition.* Washington, DC: Thompson Books, 1968.

[3] T. M. Cover and T. J. Wagner, "Topics in statistical pattern recognition," in *Digital Pattern Recognition*, (K. S. Fu, ed.), New York: Springer–Verlag, 1976.

[4] A. Pathak and S. K. Pal, "A generalised learning algorithm based on guard zones," *Pattern Recognition Letters*, vol. 4, no. 2, pp. 63–69, 1986.

[5] A. Pathak and S. K. Pal, "Learning with mislabelled training samples using stochastic approximation," *IEEE Transactions on Systems, Man and Cybernatics*, vol. SMC-17, no. 6, pp. 1072–1077, 1987.

[6] A. Pal (Pathak) and S. K. Pal, "Effect of wrong samples on the convergence of learning processes,". To be published in *Information Sciences.*

[7] A. Pal (Pathak) and S. K. Pal, "Effect of wrong samples on the convergence of learning processes - ii : a remedy,". To be published in *Information Sciences.*

[8] S. K. Pal, A. Pathak, and C. Basu, "Dynamic guard zone for self-supervised learning," *Pattern Recognition Letters*, vol. 7, pp. 135–144, 1988.

[9] A. Pal, "Optimum thresholds for a class of learning algorithms," in *Electronic Circuits and Systems: Proceedings of the National Conference on Electronic Circuits and Systems, Roorkee, India*, (Roorkee, India), pp. 443–445, Bombay: Tata McGraw-Hill, Nov 2-4 1989.

[10] A. Pal (Pathak) and S. K. Pal, "Generalized guard zone algorithm (GGA) for learning : automatic selection of threshold,". To be published in *Pattern Recognition.*

[11] R. R. Bush and F. Mosteller, *Stochastic Models for Learning.* New York: Wiley, 1958.

[12] M. Iosifescu and R. Theodorescu, *Random Processes and Learning.* New York: Springer-Verlag, 1969.

[13] M. F. Norman, *Markov Processes and Learning Models.* Academic Press, 1972.

[14] S. Lakshmivarahan, *Learning Algorithms Theory and Applications.* New York: Springer-Verlag, 1981.

[15] Y. Z. Tsypkin, *Foundations of the Theory of Learning Systems.* New York: Academic Press, 1973.

[16] K. S. Fu, "Learning control systems - review and outlook," *IEEE Transactions on Automatic Control*, vol. AC-15, pp. 210–221, 1970.

[17] K. S. Fu, "Learning control systems and intelligent control systems: intersection of artificial intelligence and automatic control," *IEEE Trans. Automatic Control*, vol. 16, pp. 70–72, 1971.

[18] R. Solomonoff, "Some recent work in artificial intelligence," *Proc. IEEE*, vol. 54, pp. 1687–1697, 1966.

[19] K. Fukunaga, *Introduction to Statistical Pattern Recognition.* New York: Academic Press, 1972.

[20] K. S. Fu, *Pattern Recognition and Machine Learning.* New York: Plenum Press, 1971.

[21] N. J. Nilsson, *Problem-solving Methods in Artificial Intelligence.* New York: McGraw-Hill, 1971.

[22] J. Slagle, *Artificial Intelligence and Heuristic Programming.* New York: McGraw-Hill, 1975.

[23] M. L. Tsetlin, *Automaton Theory and Modelling of Biological Systems.* New York: Academic, 1973.

[24] K. S. Narendra and M. A. L. Thathachar, "Learning automata – a survey," *IEEE Transaction on Systems, Man and Cybernetics*, vol. SMC-4, pp. 323–334, 1974.

[25] S. Grossberg, ed., *Neural Networks and Natural Intelligence.* Cambridge, Massachusetts: The MIT Press, 1988.

[26] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representations by error propagation," in *Parallel Distributed Processing: Explorations in the Microstructures of Cognition*, (D. E. Rumelhart and J. L. McClelland, eds.), Cambridge, Massachusetts: The MIT Press, 1986.

[27] J. Kittler, ed., *Pattern Recognition.* Berlin: Springer–Verlag, 1988.

[28] D. E. Rumelhart and J. L. McClelland, eds., *Parallel Distributed Processing: Explorations in the Microstructures of Cognition.* Vol. I, Cambridge, Massachusetts: The MIT Press, 1986.

[29] D. E. Rumelhart and D. Zipser, "Competitive learning," in *Parallel Distributed Processing: Explorations in the Microstructures of Cognition*, (D. E. Rumelhart and J. L. McClelland, eds.), Cambridge, Massachusetts: The MIT Press, 1986.

[30] I. Aleksander, "Adaptive pattern recognition systems and Boltzmann machines: a rapproachment," *Pattern Recognition Letters*, vol. 6, pp. 113–120, 1987.

[31] R. P. Lippmann, "An introduction to computing with neural nets," *IEEE ASSP Magazine*, vol. 4, pp. 4–22, 1987.

[32] K. S. Fu, "Learning system theory," in *System Theory*, (L. A. Zadeh and E. Polak, eds.), New York: McGraw-Hill, 1969.

[33] N. J. Nilsson, *Learning Machines*. New York: McGraw-Hill, 1965.

[34] J. Sklansky and G. N. Wassel, *Pattern Classifiers and Trainable Machines*. New York: Springer-Verlag, 1981.

[35] K. S. Fu, *Sequential Methods in Pattern Recognition and Machine Learning*. New York: Academic, 1968.

[36] K. S. Fu, "Relationships among various learning techniques in pattern recognition systems," in *Pattern Recognition*, (L. Kanal, ed.), pp. 399–408, Washington,DC: Thompson Books, 1968.

[37] R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis*. New York: Wiley, 1973.

[38] J. T. Tou and R. C. Gonzalez, *Pattern Recognition Principles*. New York: Addison–Wesley, 1974.

[39] J. A. Hartigan, *Clustering Algorithms*. New York: Wiley, 1975.

[40] P. H. A. Sneath and R. Sokal, *Numerical Taxonomy*. San Fransisco: Freeman, 1973.

[41] M. R. Anderberg, *Cluster Analysis for Applications*. New York: Academic Press, 1973.

[42] Y. T. Chien and K. S. Fu, "On bayesian learning and stochastic approximation," *IEEE Transactions on Systems Science and Cybernetics*, vol. SSC-3, pp. 28–38, 1967.

[43] M. T. Wasan, *Stochastic Approximation*. Cambridge: Cambridge University Press, 1969.

[44] M. B. Nevelson and R. Z. Has'minskii, *Stochastic Approximation and Recursive Estimation*. Providence, Rhode Island: American Mathematical Society, 1973.

[45] H. Robbins and S. Monro, "A stochastic approximation method," *Annals of Mathematical Statistics*, vol. 22, pp. 400–407, 1951.

[46] J. R. Blum, "Multidimensional stochastic approximation methods," *Annals of Mathematical Statistics*, vol. 25, pp. 382–386, 1954.

[47] A. Dvoretzky, "On stochastic approximation," in *Proceedings of the 3rd Berkeley Symposium on Mathematical Statistics and Probability*, (J. Neyman, ed.), (Berkeley, California), pp. 39–55, University of California Press, 1956.

[48] J. Kiefer and J. Wolfowitz, "Stochastic estimation of the maximum of a regression function," *Annals of Mathematical Statistics*, vol. 23, pp. 462–466, 1952.

[49] A. E. Albert and L. A. Gardner, *Stochastic Approximation and Nonlinear Regression*. Cambridge, MA: MIT Press, 1967.

[50] R. Rosenblatt, *Principles of Neurodynamics*. New York: Spartan Books, 1959.

[51] M. Minsky and S. Papert, *Perceptrons: An Introduction to Computational Geometry*. Cambridge, Massachusetts: The MIT Press, 1969.

[52] B. Widrow and M. E. Hoff, "Adaptive switching circuits," Tech. Rep. TR 1553-1, Stanford University, Stanford, California, 1960.

[53] S. K. Pal, A. K. Dutta, and D. D. Majumder, "A self-supervised vowel recognition system," *Pattern Recognition*, vol. 12, pp. 27–34, 1980.

[54] Y. T. Chien, "The threshold effect of a non-linear learning algorithms for pattern recognition," *Information Science*, vol. 2, pp. 351–358, 1970.

[55] D. F. Andrews, *Robust Estimates of Location : Survey and Advances*. Princeton, NJ: Princeton University Press, 1972.

[56] P. J. Huber, *Robust Statistics*. New York: Wiley, 1981.

[57] F. R. Hampel, E. M. Ronchetti, P. J. Rousseeuw, and W. A. Stahel, *Robust Statistics: The Approach based on Influence Functions*. New York: John Wiley, 1986.

[58] V. Barnett and T. Lewis, *Outliers in Statistical Data*. New York: Wiley, 1978.

[59] W. J. J. Rey, *Introduction to Robust and Quasi-Robust Statistical Methods*. Springer-Verlag, 1983.

[60] C. R. Rao, *Linear Statistical Inference and Applications*. Wiley, 1973.

[61] R. S. Chhikkara and J. McKeon, "Linear discriminant analysis with misallocation in training samples," *Journal of the American Statistical Association*, vol. 79, pp. 899–906, 1984.

[62] C. B. Chittineni, "Estimation of probabilities of label imperfections and correction of mislabels," *Pattern recognition*, vol. 13, pp. 257–268, 1981.

[63] P. A. Lachenbruch, "Discriminant functions when the initial samples are misclassified," *Technometrics*, vol. 8, pp. 657–652, 1966.

[64] P. A. Lachenbruch, "Discriminant functions when the initial samples are misclassified II : nonrandom misclassification models," *Technometrics*, vol. 16, pp. 419–424, 1974.

[65] D. M. Titterington, "Updating a diagnostic system using unconfirmed cases," *Applied Statistics*, vol. 25, pp. 238–247, 1976.

[66] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood with incomplete data via the EM algorithm," *Journal of the Royal Statistical Society*, vol. 39B, pp. 1–38, 1977.

[67] G. J. McLachlan, "Estimating the linear discriminant function from initial samples containing a small number of unclassified observations," *Journal of the American Statistical Association*, vol. 72, pp. 403–406, 1977.

[68] J. E. Michalek and R. C. Tripathi, "The effect of errors in diagnosis and measurement on the estimation of probability of an event," *Journal of the American Statistical Association*, vol. 75, pp. 713–721, 1980.

[69] T. J. O'Neill, "Normal discrimination with unclassified observations," *Journal of the American Statistical Association*, vol. 73, pp. 821–826, 1978.

[70] T. Krishnan, "Efficiency of normal discrimination with misclassified initial samples," Tech. Rep. ASC/85/3, Indian Statistical Institute, Calcutta, 1985.

[71] T. Krishnan, "Efficiency of learning with imperfect supervision," *Pattern Recognition*, vol. 21, pp. 183–188, 1988.

[72] U. A. Katre and T. Krishnan, "Pattern recognition with an imperfect teacher," in *Pattern Recognition in Practice*, (E. S. Gelsema and L. N. Kanal, eds.), Amsterdam: North Holland, 1985.

[73] T. Krishnan, "Role of supervisor in learning: a survey," Tech. Rep. ASC/85/5, Indian Statistical Institute, Calcutta, India, 1985.

[74] W. Greblicki, "Learning to recognize pattern with a probabilistic teacher," *Pattern Recognition*, vol. 12, pp. 159–164, 1980.

[75] T. Krishnan and S. C. Nandy, "Discriminant analysis with a stochastic supervisor," *Pattern Recognition*, vol. 20, pp. 379–384, 1987.

[76] D. M. Titterington, "An alternative stochastic supervisor in discriminant analysis," *Pattern Recognition*, vol. 22, pp. 91–95, 1989.

[77] C. B. Chittineni, "Learning with imperfectly labeled samples," *Pattern Recognition*, vol. 12, pp. 281–291, 1980.

[78] D. C. Farden, "Stochastic approximation with correlated data," *IEEE Transactions on Information Theory*, vol. IT-27, pp. 105–113, 1981.

[79] D. M. Young, D. W. Turner, and V. R. Margo, "On the robustness of the equal-mean discrimination rule with uniform covariance structure against serially correlated training data," *Pattern Recognition*, vol. 21, no. 2, pp. 189–194, 1988.

[80] C. R. O. Lawoko and G. J. McLachlan, "Further results on discrimination with autocorrelated observations," *Pattern Recognition*, vol. 21, no. 1, pp. 69–72, 1988.

[81] L. Schmetterer, "Multidimensional stochastic approximation," in *Multivariate Analysis : Proc. 2nd Int. Symp. Multivariate Analysis*, (P. R. Krishnaiah, ed.), New York: Academic, 1968.

[82] C. R. Rao and S. K. Mitra, *Generalized Inverse of Matrices and Its Applications.* New York: Wiley, 1971.

[83] K. Knopp, *Theory and Application of Infinite Series.* Glasgow: Blackie, 1959.

[84] K. Pearson, *Tables of the Incomplete Beta Function.* London: Biometrika Trustees, 1968.

[85] H. Wold, *Random Normal Deviates. Tracts for Computers*, Cambridge: University Press, 1954.

[86] A. K. Datta, N. R. Ganguli, and S. Ray, "Recognition of unaspirated plosives–a statistical approach," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. ASSP-28, pp. 85–91, 1980.

[87] S. K. Pal, *Studies on the Application of Fuzzy Set Theoretic Approaches in Some Problems of Pattern Recognition and Man-Machine Communication by Voice.* PhD thesis, Calcutta University, 1978.

[88] D. D. Majumder *et al.*, "Application of pattern recognition and image processing techniques to geological mapping and mineral detection," Tech. Rep., Electronics and Communication Sciences Unit, Indian Statistical Institute, Calcutta, India, 1989.

[89] S. K. Pal, "Optimum guard zone for selfsupervised learning," *IEE proceedings*, vol. 129, pp. 9–14, 1982.

[90] Y. T. Chien, "Linear and nonlinear stochastic approximation algorithms for learning systems," in *Pattern Recognition and Machine Learning*, (K. S. Fu, ed.), New York: Plenum Press, 1971.

[91] H. E. Stephanou, "Imperfectly supervised classification using fuzzy prototypes," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Fransisco, California*, New York: IEEE Computer Society Press/North Holland, 1985.

[92] Y. T. Chien, *Interactive Pattern Recognition.* New York: Marcel Dekker, 1978.

LIST OF PUBLICATIONS
of the author

1. Syntactic Recognition of Skeletal Maturity, *Pattern Recognition Letters*, Vol. 2, pp. 193–197, 1984.

   Co-authors: S. K. Pal and R. A. King

2. Learning of a Speech Recognition System using Stochastic Approximation, *Proc. IEEE Int. Conf. Comp., Syst. and Signal Processing*, Bangalore, India, Dec. 9–12, 1984, pp. 910–913.

   Co-author: S. K. Pal

3. Fuzzy Approach to the Syntactic Recognition of Skeletal Maturity, *Proc. IEEE Int. Conf. Comp., Syst. and Signal Processing*, Bangalore, India, Dec. 9–12, 1984, pp. 58–62.

   Co-author: S. K. Pal

4. A Generalised Learning Algorithm Based on Guard Zones, *Pattern Recognition Letters*, Vol. 4, No. 2, pp. 63–69, 1986.

   Co-author: S. K. Pal

5. Fuzzy Grammars in Syntactic Recognition of Skeletal Maturity from X-Rays, *IEEE Trans. Syst., Man and Cyberns.*, Vol. 16, No. 5, pp. 657–667, 1986.

   Co-author: S. K. Pal

6. On the Convergence of 'A Self-supervised Vowel Recognition System', *Pattern Recognition*, Vol. 20, No. 2, pp. 237–244, 1987.

   Co-author: S. K. Pal

7. Learning with Mislabelled Training Samples using Stochastic Approximation, *IEEE Trans. Syst., Man and Cyberns.*, Vol. 17, No. 6, pp. 1072–1077, 1987.

   Co-author: S. K. Pal

8. Dynamic Guard Zone for Self-supervised Learning, *Pattern Recognition Letters*, Vol. 7, pp. 135–144, 1988.

   Co-authors: S. K. Pal and C. Basu

9. Effect of Wrong Samples on the Convergence of Learning Processes, *Information Sciences*, (accepted for publication).

   Co-author: S. K. Pal

10. Effect of Wrong Samples on the Convergence of Learning Processes – II : A Remedy, *Information Sciences*, (accepted for publication).

    Co-author: S. K. Pal

11. Generalized Guard Zone Algorithm (GGA) for Learning : Automatic Selection of Threshold, *Pattern Recognition*, (accepted for publication).

    Co-author: S. K. Pal

12. Optimum Thresholds for a Class of Learning Algorithms, in *Electronic Circuits and Systems: Proc. Nat. Conf. Elect. Circuits and systems*, Roorkee, India, Nov.2–4, 1989.
